

Universidad Complutense de Madrid

Facultad de Informática



**Definición, desarrollo e implementación de un sistema
experto de decisión para entornos de modelado e-health**

Alumno **Fernando Viñas Martins**

Directores de proyecto **José Luis Risco Martin**
Josué Pagán Ortiz

Trabajo de Fin de Grado

Grado en Ingeniería informática

Junio de 2017

Índice general

Lista de figuras	5
Lista de tablas	7
Palabras clave	9
Resumen	11
1. Introducción	13
1.1. eHealt, IoT y la predicción de migrañas	14
1.2. Objetivos y plan de trabajo	15
2. Introduction	17
2.1. eHealt, IoT and migraine prediction	17
2.2. Targets and work plan	18
3. Estado del arte	21
3.1. Respuestas imprecisas a la predicción de migrañas	21
3.2. Aplicaciones de referencia en el ámbito móvil	23
3.3. Respuesta robusta a la predicción de migrañas	24
4. Introducción teórica	27
4.1. Aspectos de aprendizaje automático y Active Learning	27
4.1.1. Aprendizaje supervisado	27
4.1.2. Aprendizaje no supervisado	29
4.1.3. Aprendizaje semi-supervisado	29
4.1.4. Active learning	30
4.2. Algoritmos supervisados utilizados	31
4.2.1. Red ANFIS	32
4.2.2. Perceptrón multicapa	32
4.2.3. C4.5	33
5. Metodología	35
5.1. Sistema de predicción de migrañas basada en datos biomédicos	36

5.2. Sistema decisor biomédico	37
5.3. Sistema de predicción bajo variables ambientales	39
5.4. Sistema experto SSD basado en Active Learning	41
6. Fase experimental	43
6.1. Obtención y tratamiento de datos	43
6.1.1. Datos biomédicos	43
6.1.2. Datos ambientales	45
6.1.3. Datos prodrómicos y de actividad	47
6.2. Procedimiento experimental	49
7. Resultados	53
7.1. Resultados del decisor biomédico	53
7.2. Resultados del predictor ambiental	54
7.3. Resultados del sistema experto	57
8. Conclusiones y trabajo futuro	61
9. Conclusions and future work	65
Bibliografía	70
Agradecimientos	71
Autorización de difusión	73

Índice de figuras

1.1. Aspectos SSD	13
1.2. eHealt e IoT	15
3.1. Falso positivo umbral	24
3.2. Curvas migraña	25
4.1. Distribución entrenamiento-test	28
4.2. Esquema funcionamiento AL	31
4.3. Curva Entropy Sampling	31
5.1. Módulos SSD	35
5.2. E/S caja negra	36
5.3. E/S decisor	37
5.4. Adelantamiento/retraso biomédico.	38
5.5. ES ambiental	40
5.6. Interacción M-U JCLAL	42
5.7. Gráficas JCLAL	42
6.1. Curvas gaussianas	44
6.2. Pulso de dolor real	44
6.3. Curvas 0-100	46
6.4. Diagrama de flujo predictor ambiental	50
6.5. Diagrama de flujo Systems Decisor Experto	51

Índice de tablas

7.1. Res. decisor biomédico	54
7.2. Res. predictor ambiental 2014-2017	54
7.3. Matriz confusión entrenamiento 2014-2017	55
7.4. Matriz confusión test 2014-2017	55
7.5. Res. predictor ambiental 2014	55
7.6. Matriz confusión entrenamiento 2014	56
7.7. Matriz confusión test 2014	56
7.8. Res. predictor ambiental 2017	56
7.9. Matriz confusión entrenamiento 2017	57
7.10. Matriz confusión test 2017	57
7.11. Res. sistema experto experimento 1	58
7.12. Res. sistema experto experimento 2	59
7.13. Res. sistema experto experimento 3	60
7.14. Res. sistema experto experimento 4	60
7.15. Res. sistema experto experimento 5	60

Palabras clave

Palabras clave en Español

- Modelado y simulación
- Sistema experto decisor
- Perceptrón multicapa
- Adaptive neuro fuzzy inference system (ANFIS)
- Active Learning

Keywords in English

- Modeling and simulation
- Expert decision system
- Multi-layer perceptron
- Adaptive neuro fuzzy inference system (ANFIS)
- Active Learning

Resumen

Resumen en Español

Buscamos predecir crisis de migraña mediante técnicas de aprendizaje automático. Estas técnicas serán introducidas en un dispositivo electrónico que monitoriza y predice las migrañas del paciente para su uso médico.

A día de hoy, han surgido diversos enfoques sobre la predicción de migrañas que han respondido al problema. Uno ha dado muy buenos resultados. Se basa en el entrenamiento de un modelo de aprendizaje automático construido mediante variables hemodinámicas y la intensidad del dolor sufrida por el paciente. Como criterio de decisión se empleó un módulo muy sencillo. Se trata de un umbral que detecta si hay migraña. Sin embargo, debido a los posibles errores en la señal derivados de su uso en la vida real, el umbral puede dar falsos positivos al ser superado sin haber episodios de migraña. Se hace necesaria la mejora de este criterio de decisión.

El siguiente trabajo pretende abordar el desarrollo de un sistema de apoyo a decisión para mejorar la respuesta del criterio basado en umbral y sustituirlo. Plantea el desarrollo de un sistema decisor previo basado en datos biomédicos, el desarrollo de un sistema predictor de migrañas basado en datos ambientales y su integración en un sistema decisor experto mayor que incluya información acerca de los síntomas prodrómicos y de los datos de actividad para enriquecerlo.

El experimento final nos ha dado como resultado que quien comanda la robustez del sistema es el sistema decisor previo basado en datos biomédicos. Por otro lado, la influencia de los datos ambientales según el enfoque utilizado es nula, la de los datos prodrómicos es muy débil y la de los datos de actividad de los pacientes es algo más relevante. Por ello, la información periférica añadida no nos sirve de mucho.

Summary in English

The aim of this work is to predict migraine crisis with machine learning techniques. These techniques will be implemented into an electronic device, which will be used for monitoring and prediction of these migraine crisis.

Nowadays, there are several approaches about migraine prediction that have tackled the problem. One of them have offered good results. It is based on the training of a machine learning model built with hemodynamic variables and the intensity of the pain suffered by the patient. As decision criteria a simple module was used. It detects if a migraine occur through a threshold. But, because of the possible errors in the signal in a real time scenario, the threshold could give

false positives if it is reached without migraine episodes. The improvement of this decision criteria is necessary.

Additionally, this work seeks to approach the development of a Decision Support System based on environmental data to improve the response of the threshold-based criteria and to substitute it. We have planned the development of a migraine decision system based on biomedical data and the development of a migraine predictor system based on environmental data to integrate them in a mayor expert decision system, which include information about prodromic symptoms and activity data to enrich it.

The final experiment gave us the result that who commands the robustness of the final system is the previous decision system based on biomedical data. On the other hand, according to our approach, the ambient data influence is null, the influence of prodromic data is very weak and activity data is a bit more relevant. Therefore, the peripheral information added doesn't serve us much.

Capítulo 1

Introducción

Los episodios de ataque de migraña son fuertemente incapacitantes, causando un gran impacto social y económico. Son sufridos por un 10 % de la población mundial [1], suponen cuantiosos costes para los sistemas sanitarios del mundo en cuestión de atención en urgencias y de despliegue médico general en su atención (en concreto, 1222 euros por paciente al año en toda Europa) [1] y suponen un problema de productividad del trabajador o trabajadora que los sufre en su puesto de trabajo [1]. La monitorización y prevención mediante predicción de los episodios de migraña pueden suponer una mejora en la calidad de vida de toda esa masa de población, incidiendo en la reducción de los costes de tipo sanitario que puedan causar y en la mejora de su productividad. Esto justifica sólidamente la necesidad de la predicción de dichos episodios.

Contamos con un sistema predictor de migrañas construido mediante variables hemodinámicas y mediante la intensidad de dolor del paciente que nos da una respuesta muy robusta [1]. Sin embargo, el sistema de decisión utilizado está basado en umbral y es muy simple, siendo sensible a falsos positivos [1]. Hemos planteado que en lugar de usar un sistema tan simple, construiremos un sistema de soporte a decisiones.

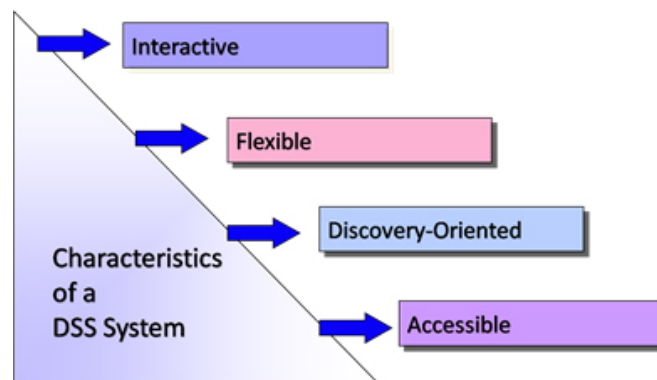


Figura 1.1: Aspectos de los SSD.

Un Sistema de Soporte a Decisiones (SSD o DSS en sus siglas en inglés)¹, es un sistema

¹https://es.wikipedia.org/wiki/Sistemas_de_soporte_a_decisiones

informático interactivo de decisión que se apoya en métodos de minería de datos para conseguir modelos de un determinado proceso estudiado de la realidad. Procurará ayudar, sin necesariamente sustituir (pues no automatizan el proceso), mediante las diversas estimaciones que pueda realizar dicho sistema, a la toma de decisiones que pueda realizar un operador ante una tarea determinada. En última instancia, se tratan de sistemas que requieren de ayuda humana, por lo que deben ser necesariamente entornos amigables, flexibles, interactivos y orientados a la experimentación y el descubrimiento sobre el proceso estudiado, como podemos ver en la siguiente figura (ver Figura 1²), para el operador que los emplee. Los ámbitos de aplicación que cubre son muy diversos, desde ámbitos económicos³ y científicos a nivel general así como ámbitos de carácter médico.

Nuestra empresa se enfrenta a la tarea de desarrollar un sistema de ayuda a la predicción para dar soporte a un sistema predictor de migrañas construido mediante variables hemodinámicas y la intensidad de dolor del paciente.

1.1. eHealth, IoT y la predicción de migrañas

Como hemos comentado anteriormente, en especial, nos interesa la aplicación de los SSD en el ámbito de la salud, y en concreto, en el ámbito de la predicción de las migrañas.

Esta aplicación se encuadra en lo que se denomina como eHealth, un tipo de práctica médica que combina los procedimientos médicos usuales y las tecnologías de la información actuando como apoyo⁴. Es un concepto amplio que puede conllevar la consigna en bases de datos de historiales médicos electrónicos, servicios de telemedicina, medicina basada en pruebas, y servicios basados en la comunicación entre pacientes y profesionales⁵. eHealth no sólo contempla el ámbito médico, si no que también contempla el ámbito cotidiano con usos muy generalizados, por ejemplo, el uso de pulseras electrónicas para monitorización personal⁶, monitorización remota para el cuidado de ancianos⁷, etc.

La incorporación del Internet de las Cosas (también conocido como Internet of Things o IoT) en el ámbito del eHealth ha permitido su evolución, tanto en lo que respecta a la monitorización, así como en el análisis de datos, debido al uso combinado de aparatos de extracción de datos cada vez más económicos y bases de datos en la nube, como podemos ver en la siguiente figura (ver Figura 1.2).

El uso de IoT en el campo médico busca cubrir la monitorización y tratamiento de enfermedades de carácter crónico, procurando ayudar a la labor de detección de los síntomas más graves o molestos para el paciente de una enfermedad crónica, y el diagnóstico de cualquier otro cuadro clínico para dar apoyo a las decisiones de los equipos médicos. Centrándonos en la monitorización y tratamiento de enfermedades de carácter crónico, por ejemplo, tenemos el caso de los episodios de ataques epilépticos, de los posibles períodos de niveles anormales de glucosa en sangre en un

²<http://www.fintellix.com/blog/?p=117>

³<http://www.fintellix.com/blog/?p=117>

⁴<https://es.wikipedia.org/wiki/ESalud>

⁵<https://es.wikipedia.org/wiki/ESalud>

⁶<http://terrenodeportivo.com/mejores-pulseras-de-actividad-baratas/>

⁷<http://www.aarp.org/espanol/hogar-familia/tecnologia/info-2014/monitorear-ancianos-distancia-herramientas.html>

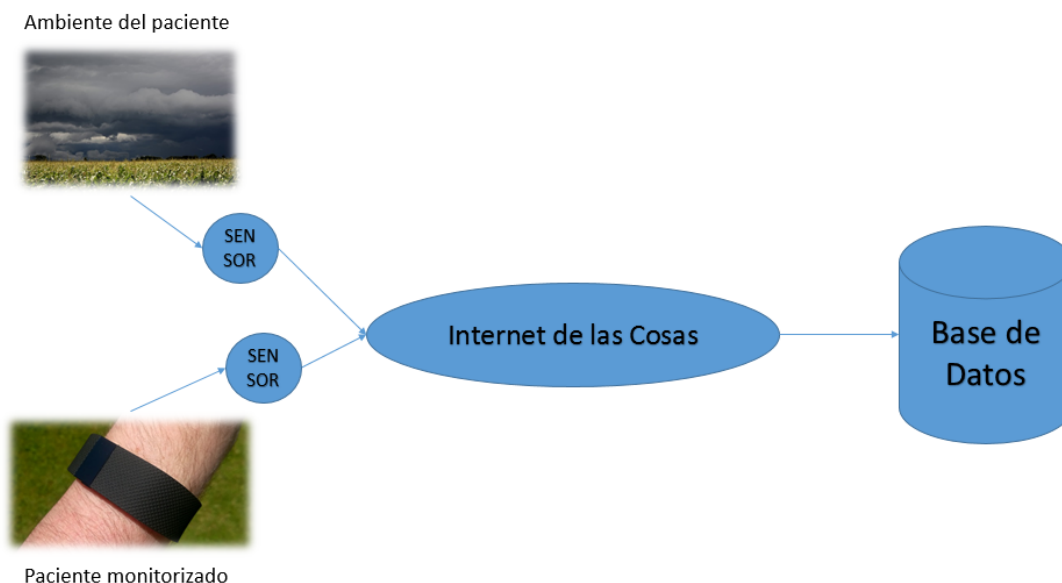


Figura 1.2: Integración de eHealth con IoT.

paciente o de los episodios de ataques de migrañas [1]. Sin embargo, en muchos casos, estos estudios y aplicaciones aún no han abordado la monitorización activa online del paciente y se han quedado en el tratamiento de bases de datos offline.

Referimos anteriormente la necesidad de predecir las crisis de migraña. En cuanto a la posibilidad de predecirlas, ha sido demostrada mediante dos vías distintas. En primer lugar, mediante el uso de pródromos y auras, que, según ciertos estudios, han demostrado ser parcialmente efectivos [2]. Y en segundo lugar, mediante la curva de intensidad de dolor entrenada en asociación a variables hemodinámicas, dando una respuesta más robusta [3]. Creemos que es posible mejorar ambas vías, combinando sus dos enfoques. Además, existen ciertos disparadores de migraña de carácter ambiental que afectan a un determinado grupo de pacientes, y que hay estudios que demuestran su incidencia [4]. Aparte, otros disparadores de actividad tienen también importancia en el desencadenamiento de migrañas [5].

1.2. Objetivos y plan de trabajo

Nuestro objetivo principal consiste en el desarrollo de un sistema de soporte a decisiones para la predicción de migrañas, que tendrá como objetivos:

- Desarrollo de un sistema predictor de migraña basado en datos ambientales mediante algoritmos de machine learning.
- Desarrollo de un sistema de decisión empleando técnicas de aprendizaje basadas en lógica difusa y entrenadas con datos biomédicos.

- Desarrollo de un sistema decisor experto mayor que genere una alarma que notifique al paciente de la llegada de una crisis de migraña. Este sistema se implementa mediante técnicas de Active Learning y se basa en información biomédica y ambiental, obtenida de los dos sistemas previos, y de información prodrómica y de actividad de los pacientes.

Las fases de trabajo que guiarán a nuestros resultados serán:

- Construcción de un sistema offline, entrenado mediante una base de datos, que aborde el problema mediante un modelo basado principalmente en Active Learning.
- Demostración de si es posible la mejora del enfoque basado en variables biomédicas y datos hemodinámicos.

El modelo de la primera fase será un primer acercamiento en el tratamiento del problema, prototipando un futuro sistema hardware puramente automático cuyo funcionamiento será fundamentalmente online. El sistema offline nos permite simplificar el desarrollo de alternativas de cara al sistema final, que incorporará un dispositivo de monitorización ambulatoria capaz de extraer datos del paciente y tratarlos de forma remota [1].

Capítulo 2

Introduction

Migraine crisis are very incapacitating episodes, generating a great social and economic impact. Is suffered by 10 % of worldwide population, introduce expensive costs into the health systems of the world in Urgencies attention and medical general care (specifically, 1222 euros per patient a year in all Europe), and generates a productivity problem of the workers that suffer it in their job. Monitoring and prevention of migraine crisis episodes prediction could suppose an improvement in live quality of that mass of population, influencing on the reduction of the health system costs and the improvement of job productivity of migraine sufferers workers. This solidly justifies the need for the prediction of such episodes.

We have a migraine prediction system built with hemodynamic variables and the pain intensity of the patient with a robust response. However, the used decision system is based on threshold and it's very simple, being sensitive to false positives. We planned that instead of using that simple system, we will develop a Support Decision System.

A Support Decision System is an interactive computer decision system that is based on data mining methods to obtain models of a studied process of reality. It will help, without substitution (because it doesn't automatize the process), with the estimations. The same estimations that could make a human operator facing the same task. In last instance, they are systems that require human intervention, so SSDs must be friendly environments, flexible, interactive and experiment-discovery oriented for the user. Their application fields are diverse: economic uses, scientific uses and also medical uses.

Our enterprise face the task of the development of a system to help prediction to give support to a previous block that predicts migraine crisis with biomedical variables.

2.1. eHealt, IoT and migraine prediction

As we have referred previously, we are especially interested in its health application, specifically, in the field of the migraine prediction.

This application fits in the field of eHealth, a kind of medical practice that combines usual medical procedures and information technologies as a support. It is a large concept that could allow the preservation of electronic medical histories in databases, telemedicine services, medicine based in evidence, and services based in communications patient-medical groups. eHealth also

looks other fields than medical field. It looks also daily use, like the use of wearable bracelet for personal monitoring, monitoring for senior care, etc.

The incorporation of Internet Of Things (also know as IoT) to the field of eHealth allowed its evolution, both in terms of monitoring as well as in data analysis, due to the combined use of increasingly economical data extraction devices and cloud-based databases.

The use of IoT on medical field seeks the coverage of monitoring and treatment of chronic diseases, looking for help to the labor of detection of the more serious or annoying symptoms from the patient of a chronic disease, and the diagnosis of another clinical presentation to give support to the decisions of medical groups. Focusing in the monitoring and treatment of chronic diseases, for example, we have the case of episodes of epileptic attacks, the case of possible periods of abnormal levels of glucose in blood, and the migraine crisis episode. But, in a lot of cases, these studies and applications did not yet approached the active monitoring of the patient and its approach only accomplished the treatment of offline databases.

We referred previously the necessity of migraine prediction. About the possibility of prediction, it have been demonstrated with two different ways. In first place, with the use of prodromes and auras, which, according to some studies, demonstrate been partially effective. In second place, with pain intensity curves trained with hemodynamic variables, giving a more robust response. We believe that its possible improve both, combining the two approaches. Also, there are environmental triggers that impact in a certain group of patients, and so many studies demonstrate their incidence. Also, other activity triggers have importance in migraine triggering.

2.2. Targets and work plan

Our main target is to develop a SSD for migraine prediction, that will have the following objectives:

- The development of a migraine prediction system based on environmental data with machine learning algorithms.
- The development of a decision system using learning techniques based on fuzzy logic and trained with biomedical data.
- The development of an expert decision system that generates an alarm that notifies to the patient the incoming of a migraine crisis. This problem is implemented with Active Learning techniques and is based in biomedical and environmental information from the two previous systems, and prodromic and activity information from the patients.

The two phases that will guide our result will be:

- The construction of an offline system, trained with a database, that will approach the problem, with a model mainly based on Active Learning.
- The proof that the improvement of the approach based in biomedical variables and hemodynamic data is possible.

The model of the first phase will be a first approach in the treatment of the problem, prototyping a future “live” and automatic hardware system. The offline system simplifies the development of alternatives for the final actual system, which will incorporate an ambulatory monitoring device able to extract data from the patient and treat them remotely.

Capítulo 3

Estado del arte

Los sistemas de soporte a decisiones¹ son capaces de dotar a un investigador de un entorno informático interactivo y adaptable a sus necesidades que le permita realizar la predicción de algún posible episodio de una enfermedad crónica o el diagnóstico de cualquier otro cuadro clínico. Les brindará como apoyo sistemas de aprendizaje automático capaces de ofrecerles estimaciones diversas sobre los que apoyar sus decisiones. Sin embargo, adolecen de la imposibilidad de automatizar el proceso al 100 %² y de la posibilidad de que en sus estimaciones pueda darse lugar a fallos, haciendo necesaria la supervisión humana. Este problema es inevitable, pues los SSD construyen hipótesis con base en los conocimientos introducidos antes del tratamiento del problema, dándoles una visión sesgada. Sin embargo, nos valen debido a que pueden suponer un fuerte ahorro en costes de atención y monitorización médica.

Existen otras iniciativas de investigación muy recientes que buscan ahondar en el campo de la predicción de episodios de dolencias crónicas, como en nuestro caso. En concreto, una de la Universidad Autónoma de Madrid, que se encuentra referida en el artículo siguiente³.

3.1. Respuestas imprecisas a la predicción de migrañas

Las primeras respuestas a la cuestión de la predicción de migrañas se han fundamentado en el enfoque basado en los síntomas del período prodrómico y en los síntomas propiciados por el aura ocasional previa a un episodio de dolor de migraña [2]. Debido a la ocasionalidad de aparición del aura y de la alta imprecisión cualitativa y temporal del prodrómico, se ha encontrado que existen serias limitaciones para que este enfoque en exclusiva sea fructífero y dé una predicción lo suficientemente consistente y satisfactoria del episodio de dolor [1]. El rango de confianza hacia el futuro del horizonte de predicción se encuentra entre las 4 horas y las 48 horas después de haber realizado la predicción [1]. Esta migraña predicha puede desencadenarse en cualquier momento de ese rango temporal, lo que hace al enfoque inútil para un uso médico. Esto no quiere decir que el enfoque no ayude a la predicción si se combina con otras formas de predicción.

Por ejemplo, en este artículo [1], se presenta un sistema de predicción donde el paciente porta

¹https://es.wikipedia.org/wiki/Sistemas_de_soporte_a_decisiones

²https://es.wikipedia.org/wiki/Sistemas_de_soporte_a_decisiones

³<http://www.iic.uam.es/lasalud/promesas-realidad-del-big-data-salud-ehealth-week/>

un dispositivo que recoge los valores de cuatro variables hemodinámicas. Los autores generan un modelo analítico que predice la aparición de una crisis migrañosa hasta con 30 minutos de antelación en promedio, aunque pueden aparecer falsos positivos y negativos. Los autores afirman por tanto dos máximas que se abordan en este trabajo: (i) el sistema desarrollado carece de un decisor que elimine falsos positivos o falsos negativos, y (ii) cualquier información extra que ayude al decisor tendría especial valor en este ámbito.

Respecto al segundo punto mencionado anteriormente, se ha reportado que cierto porcentaje de la población migrañosa en todo el mundo afirma sufrir episodios de dolor a causa de diversas variaciones en las condiciones meteorológicas en las que se haya inmerso, funcionando a modo de disparador [4]. Esto incluye tanto las variaciones ambientales en el exterior, más fáciles de monitorizar, como las variaciones ambientales relacionadas con espacios interiores, más difíciles de monitorizar [4]. Si bien existen estudios científicos que corroboran con fuerza que las condiciones meteorológicas afectan a dicho porcentaje de pacientes [4], las causas, las formas de interacción y los factores concretos aún se encuentran en controversia dentro de la comunidad científica [4]. Diversos trabajos estadísticos sí han conseguido encontrar una correlación real y consistente entre los cambios meteorológicos y el dolor de ciertos pacientes. Sin embargo, a la hora de elucidar las causas que pueden haber producido dichas migrañas a través de las preguntas de los pacientes, cualquier posible estructuralidad particular entre sus apreciaciones subjetivas termina por derrumbarse [4]. Estudios recientes afirman que los pacientes enfermos de migraña son más sensibles y sienten molestias ante la exposición a la luz, ante perturbaciones sonoras y olfativas, y a ilusiones y distorsiones en forma de grilla de tipo sinusoidal, por lo que sus limitaciones perceptivas pueden tener que ver en esta cuestión [4]. Esta información puede darnos una ayuda en la predicción de la migraña, como en el caso de los datos prodrómicos, información que se ha añadido en la implementación realizada en este trabajo.

Asimismo, se ha planteado que puede resultar de utilidad la recogida de información cualitativa de toda eventualidad que pueda funcionar como disparador para las migrañas mediante aplicaciones móviles tipo calendario, consiguiendo predecir un episodio de dolor [5]. Dicha información estaría conformada tanto por las diversas actividades realizadas por el paciente, como por los ritmos biológicos que puedan influir, aspectos de carácter alimenticio, etc., que el paciente debe cumplimentar en un formulario para ser guardadas en una base de datos remota. Estas aplicaciones, idealmente, podrían dar consejos al paciente para evitar episodios de migraña, e información cualitativa a los profesionales médicos especializados en la materia. Puede legarnos una base de datos extensa sobre la que aplicar métodos de minería de datos y extraer conclusiones interesantes sobre dicha información para un sistema de aprendizaje automático. Sin embargo, según un estudio del 2014 [5], de las 38 aplicaciones estudiadas ninguna cumplía con todos los requisitos mínimos para hacerla aceptable según su criterio. De ellas destacaban tres que cumplían casi todos: iHeadache, echoHeadache y Headache Diary Pro. Sin embargo, el estudio lamentaba en sus conclusiones que la base científica y el apoyo y asesoramiento experto de la mayor parte de las aplicaciones son escasos, sólo figurando en el 18 % de las aplicaciones [5]. Sin embargo, las predicciones realizadas, si bien son válidas en tanto que detectan si habrá migraña, no determinan bien el horizonte de predicción y, por lo tanto, no sabemos cuándo puede ocurrir con una precisión

confiable. A continuación, se repasan algunas de las aplicaciones más relevantes para registrar la actividad del paciente.

3.2. Aplicaciones de referencia en el ámbito móvil

Para profundizar en las propuestas más modernas y avanzadas de aplicaciones móviles para el seguimiento de las migrañas, veremos dos de las aplicaciones de referencia en el ámbito de las tiendas de aplicaciones móviles: en primer lugar Migraine Buddie, presente en la appstore de Android, y en segundo lugar MigraineAlert, que se encuentra en la appstore de Apple.

Según afirma su página en PlayStore⁴, Migraine Buddie⁵, aparecida en 2014, es una aplicación para la datación en el tiempo y la consigna de los datos de las diversas migrañas que sufra el paciente diseñada por expertos en la materia. La funcionalidad general no es muy diferente a las aplicaciones que hemos referido anteriormente, si bien tiene un fuerte enfoque de interacción médico-paciente. Sin embargo, difiere frente a otras propuestas en que presenta en su calendario un diario del sueño inteligente que permite a los pacientes ver la relación existente entre las horas de sueño y las migrañas mediante una representación gráfica en la que, con mucha probabilidad, intervenga algún tipo de sistema de aprendizaje automático en su construcción. Además, según afirma un artículo promocional de la aplicación⁶, se vale de los sensores de movimiento del teléfono móvil en que se encuentre la aplicación para recoger tanto información sobre el movimiento del paciente como información sobre los procesos del sueño, permitiendo que, en caso de que el paciente estuviera muy aturdido como para incorporar información cualitativa y temporal de la migraña, se dispusiera de esa información.

La segunda aplicación que trataremos, MigraineAlert⁷, aparecida a finales de 2016, resulta ser una propuesta mucho más seria que la anterior. Su enfoque, que se encuentra completo en esta página⁸, se basa en la generación de hipótesis individuales construidas mediante métodos de aprendizaje automático para cada paciente. Este enfoque parte del supuesto de que, debido a que la causalidad de la migraña es múltiple y no estructural, no es posible encontrar un modelo que pueda valer para todos los pacientes. Mediante la aplicación y un aparato llamado Fitbit, que se supone que el paciente debe ponerse para realizar monitorizaciones corporales, el sistema extrae datos de tipo hemodinámico, ambiental y disparadores de síntomas y actividad, que serán almacenados en una base de datos. La información extraída sería tratada de forma diferenciada en dos fases de aprendizaje divididas con claridad en el tiempo. En la primera, que dura como máximo 75 días, el sistema sigue pasivamente los diversos eventos que vive el paciente y entrena al modelo predictivo. En la segunda fase, que dura 75 días, el modelo toma parte activa y se utilizará sobre los sucesos que se monitoricen sobre el paciente para testarlo. Para comprobar su eficacia se realizaría un análisis diferencial de las migrañas que prediga y el tiempo en que lo consiga confrontándolo con

⁴<https://play.google.com/store/apps/details?id=com.healint.migraineapp&hl=es>

⁵<http://www.migrainebuddy.com/>

⁶<http://techcrunch.com/2014/07/29/migraine-buddy/>

⁷<https://somobilehealth.com/index.html>

⁸<https://clinicaltrials.gov/ct2/show/NCT02910921?term=migraine+alert+app&rank=1>

los sucesos reportados por el paciente a través de la aplicación.

Estas aplicaciones incorporan elementos poco vistos, como Migraine Buddie con su análisis inteligente del sueño o como MigraineAlert, que hace seguimiento de algunas variables hemodinámicas y ambientales del paciente mediante el aparato Fitbit. Sin embargo, el grueso del enfoque se basa en el uso de predicciones mediante síntomas prodrómicos, lo que hacen a estas aplicaciones limitadas por el carácter impreciso de sus bases.

3.3. Respuesta robusta a la predicción de migrañas

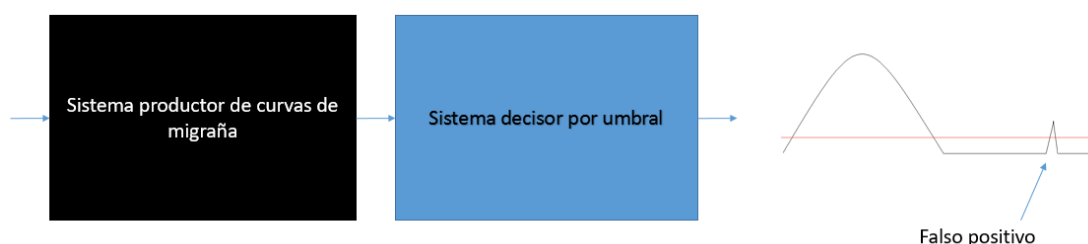


Figura 3.1: Falsos positivos captados por el umbral.

Como se ha mencionado anteriormente, hay un estudio en particular [3] que demuestra que es posible predecir un episodio de dolor de migraña mediante la monitorización de diversas variables hemodinámicas puestas en relación con datos de dolor contruidos en base a la intensidad del dolor reportada por el paciente. La Figura 3.2 nos muestra cómo se modela la curva de dolor. Para ello, se obtiene la sensación registrada por el paciente de intensidad de dolor, que se ajusta posteriormente a una curva gaussiana que servirá de base para el entrenamiento del modelo predictivo [3]. En concreto, la curva se pondrá en relación, mediante diversos modelos basados en algoritmos de espacios de estado, con cuatro variables hemodinámicas distintas obtenidas de los pacientes bajo monitorización ambulatoria, mediante una red inalábrica de sensores corporales no invasivos [3]. Los horizontes de predicción obtenidos con este enfoque son satisfactorios y consistentes, con horizontes máximos de 52 minutos y de 47 minutos en promedio. En casos peores de 30 minutos promedio también han mostrado ser sólidos [3]. Los sistemas existentes que se apoyan en este enfoque usan una política de decisión basada en la aplicación de un umbral determinado en base a un valor fijo [3]. Sin embargo, en caso de ruido excesivo en la señal recogida, era posible que dicho ruido fuese aceptado por el umbral, como se puede ver en la Figura 3.1, haciendo que el sistema detectase un falso positivo. Esto hacía al umbral poco fiable en el proceso de detección de migrañas para entornos reales. Además, el establecimiento del umbral es arbitrario, lo que sesga la respuesta del sistema. Es por ello que en este trabajo se propone la incorporación de un sistema decisor que filtre los falsos positivos y evite los falsos negativos.

Hasta ahora se han presentado distintos enfoques, que por si solos, presentan una gran diversidad de problemas. No obstante, si pensamos en un sistema de aprendizaje superior, se puede dar la posibilidad de que se prescinda de la necesidad de un umbral para el caso del sistema de pre-

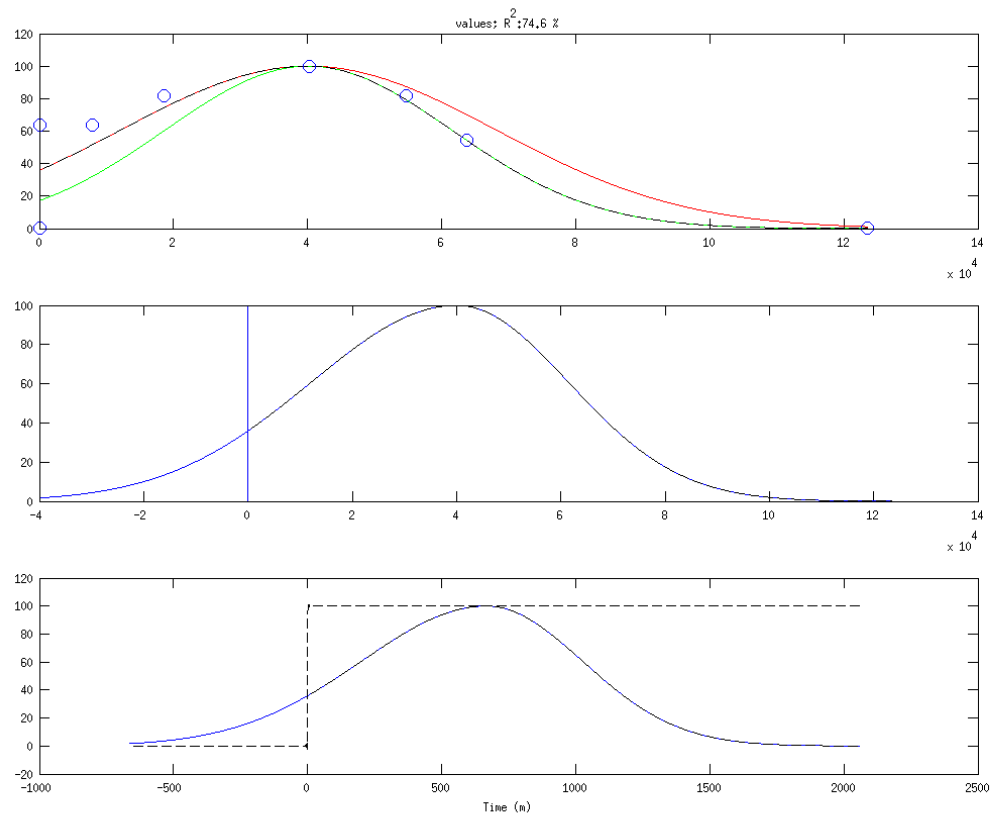


Figura 3.2: Curva gaussiana obtenida con entrenamiento mediante datos hemodinámicos.

dicción basado en variables biomédicas y construir de este modo un SSD más robusto que pueda aplicar distintas políticas de decisión en función de la influencia y precisión de cada una en cada momento. Este es el propósito último de nuestro Trabajo de Fin de Grado.

Capítulo 4

Introducción teórica

En ese capítulo se introducirán primero los conceptos teóricos que soportan la estructura de la metodología propuesta en el Capítulo 5.

4.1. Aspectos de aprendizaje automático y Active Learning

La técnica de aprendizaje automático (usaremos indistintamente “aprendizaje automático” o “machine learning”) que hemos utilizado en este proyecto se denomina *Active Learning* (en adelante, *AL*). *AL* se construye con base en tres paradigmas de aprendizaje automático: *aprendizaje supervisado*, *aprendizaje no supervisado* y *aprendizaje semi-supervisado*. *AL* sería una forma concreta del tercer paradigma.

Sobre los dos primeros daremos unas pinceladas generales. Esta vista rápida considerará principalmente la diferencia del concepto de aprendizaje supervisado frente al de no supervisado, pues encierra un conjunto de conceptos base que son fundamentales en el semi-supervisado.

4.1.1. Aprendizaje supervisado

Este paradigma de machine learning tiene como objetivo el aprendizaje de un mapa

$$h : C_1 \times \dots C_n \rightarrow S_1 \times \dots S_n \quad (4.1)$$

siendo C_i la tupla de características de entrada y S_i la de salida. Se trata de inferir una función, denominada *hipótesis*, a partir de informaciones procedentes de un conjunto de datos de entrenamiento, y que se aplicará sobre un conjunto de test. La salida puede ser tanto valores numéricos (problemas de regresión), como etiquetas de clase (problemas de clasificación) [6]. Los del segundo tipo, en concreto, nos permiten agrupar (o clusterizar, de su término en inglés) en diversas clases las diversas instancias del problema y serán los que nos interesarán en nuestro proyecto.

En el sistema propuesto, para la implementación del sistema predictor de migrañas basado en datos meteorológicos (ver Figura 5.1), se ha utilizado un sistema de aprendizaje supervisado que se detalla en la Sección 4.2. También se ha utilizado para la implementación del sistema decisor basado en datos biomédicos(ver Figura 5.1), detallándose en Sección 4.2. Además, el corazón

del esquema AL del sistema experto final (ver Figura 5.1) que hemos construido, alberga para el cálculo de su hipótesis un sistema de aprendizaje supervisado que se detalla en Sección 4.2.

El conjunto de entrenamiento que referimos lo denotaremos como

$$E = \{(x_i, y_i) | x_i \in C_1 \times \dots \times C_n, y_i \in S_1 \times \dots \times S_n\} \quad (4.2)$$

y al conjunto de test lo denotaremos como

$$P = \{(x_i, y_i) | x_i \in C_1 \times \dots \times C_n, y_i \in S_1 \times \dots \times S_n\} \quad (4.3)$$

tal que $P \cap E = \emptyset$ y ambos serán conocidos de antemano. Veremos que se distribuirán siguiendo un criterio de un 70 % para E y un 30 % para P tal como se muestra en la Figura 4.1.



Figura 4.1: Distribución de los datos entre conjunto de entrenamiento y test.

Los parámetros del algoritmo de aprendizaje los ajustaremos mediante *Hold Out* para el algoritmo empleado en el sistema AL y mediante *validación cruzada* para el predictor biomédico y el ambiental [7].

A continuación explicaremos el esquema usual de validación cruzada, llamado *de k particiones*. Dicha técnica se aplica en casos en los que la masa de datos etiquetada es muy pequeña, algo muy habitual. Esto se debe a que la baja cantidad de datos impide la posibilidad de establecer un conjunto de validación concreto, pues haría de la cantidad de datos de entrenamiento demasiado exigua [6]. Además, procurará garantizar que dichos resultados sean independientes de la partición entre el conjunto de entrenamiento y el de prueba.

Se trata de un proceso iterativo en el que se particiona aleatoriamente a la masa de datos de muestra en k particiones más o menos iguales con la siguiente forma:

$$((x_{i,1}, y_{i,1}), \dots, (x_{i,t_i}, y_{i,t_i})) \quad (4.4)$$

siendo t_i el tamaño de la partición i . Después, se toma una partición como conjunto de validación y se deja el resto como conjunto de entrenamiento. Con ambos conjuntos constituidos, se genera una hipótesis mediante la masa de datos de entrenamiento y se evalúa dicha hipótesis h_i sobre el conjunto de validación, siendo este último un conjunto distinto en cada iteración (hasta hacer k iteraciones, dato que determina el resultado generado) recorriendo la totalidad de

la masa de prueba. De los datos de error generado en la evaluación, se haría la media, a la que denominaremos *error de validación cruzada* [6].

Aparte de este esquema, utilizaremos el que *deja uno fuera*. En él se toma un elemento como entrenamiento y el resto se utiliza como validación.

El acercamiento de *Hold Out* es más simple. Sólo emplearía una partición para validar en la fase de testeo y el entrenamiento se realizaría con el conjunto de entrenamiento en su completitud. Es más sencillo, pero cuenta con la limitación de ser sesgado. Dado que lo usaremos para el sistema AL y obtener así una métrica, nos bastará [6].

4.1.2. Aprendizaje no supervisado

En este paradigma de machine learning la máquina aprendiz recibe sólo una masa de datos de entrenamiento no etiquetada y debe hacer predicciones para todos los ejemplares no conocidos previamente [7].

El objetivo del paradigma se puede resumir en la siguiente idea: siendo C un conjunto de datos no etiquetados, hay que buscar estructuras interesantes en dicho conjunto [7]. Tales estructuras conformarán la hipótesis o modelo de densidad $\mathbf{X} = (x_i^T)_{i \in [n]}^T$ [7], estimada a partir de la aplicación de alguna técnica no supervisada sobre C . Si bien no requiere de conocimiento previo, tiene como problema que el rendimiento del algoritmo puede ser muy ineficiente, y que la búsqueda para la construcción del modelo puede ser potencialmente ciega o inconsistente, por lo que este enfoque no se encontrará en nuestro caso.

4.1.3. Aprendizaje semi-supervisado

Este paradigma de machine learning procura transitar un camino intermedio entre los dos paradigmas ya vistos. Nos limitaremos a presentar el enfoque fundamentado en el paradigma supervisado *con información adicional de la distribución de los ejemplos de X* , más relacionado con nuestro proyecto. Nos centraremos en la construcción de una hipótesis h .

Aparte de información no etiquetada $X_{ne} = (x_1, \dots, x_n)$, al algoritmo que pertenece a este paradigma también se le provee con cierta información supervisada $X_e = (x_1, \dots, x_k)$, apuntando cada elemento de la tupla a los diversos elementos del conjunto $Y = (y_1, \dots, y_k)$, los cuales serán las clasificaciones o valores conocidos. Debe quedar claro que al algoritmo no se le provee necesariamente de todos los ejemplos, pues si no el enfoque tenderá a degradarse a supervisado [7].

Este paradigma de aprendizaje nos resulta interesante frente al enfoque supervisado, pues con menos conocimientos podemos obtener mejores costes, y también frente al no supervisado, pues la búsqueda estará más guiada. Sin embargo, desde el enfoque más cercano al supervisado que nosotros vamos a implementar, se ve afectado por el algoritmo supervisado que se utilice [7] como modelo de hipótesis. Además, si se introduce información de mala calidad, esta puede desinformar más que informar [7].

4.1.4. Active learning

Tomando como punto de apoyo lo desarrollado anteriormente, podemos definir a esta técnica de aprendizaje como una forma concreta de aplicación del paradigma semi-supervisado. A diferencia de los sistemas de aprendizaje de *tipo pasivo*, que inducen sólo una única hipótesis en el curso del aprendizaje, la técnica de AL comprende su construcción como una sucesión de desarrollos y pruebas constante e ininterrumpido de nuevas hipótesis, conformando así un proceso continuo e interactivo de aprendizaje. El esquema previamente mostrado del proceso de AL se acerca, en cuanto al planteamiento general, al del método científico usual [8].

En cada fase del proceso de construcción y prueba de una hipótesis nueva la técnica requiere de forma activa del suministro de nuevo conocimiento para instancias no etiquetadas existentes o sintetizadas por el propio algoritmo proveniente de un oráculo. Nosotros usaremos un oráculo humano [8], esto es, un agente externo que introducirá las diversas instancias sobre las que el sistema AL tenga dudas seleccionadas mediante un determinado criterio, que veremos más adelante.

De entre los escenarios disponibles, en nuestro proyecto hemos escogido el conocido como *Pool-Based Sampling*, o muestreo basado en encuesta. Su premisa parte de la idea de que se tiene una pequeña masa de datos etiquetada L y una gran masa de información no etiquetada, o *pool* U , disponible [8].

El acercamiento, condensado en la figura (ver Figura 4.2) se basa en el siguiente procedimiento: la máquina aprendiz, a partir de un determinado criterio selecciona el ejemplar más problemático del pool, establece la consulta $\langle x, ? \rangle$ al oráculo y, tras haber sido respondida por éste, el ejemplar recién etiquetado $\langle x, y \rangle$ pasa a formar parte del conjunto etiquetados L , probándose con el nuevo conjunto de etiquetados $L_i = L_{i-1} \cup \{\langle x, y \rangle\}$ el algoritmo de aprendizaje. Este proceso se itera sucesivamente [8].

Mediante el criterio de selección el algoritmo evalúa y jerarquiza los ejemplares no etiquetados, sacándolos en un orden que podemos presumir que es eficiente y útil a la hora de informar al proceso de aprendizaje [8].

El criterio de selección de ejemplar que utilizaremos es un subtipo del conjunto *Uncertainty Sampling Strategy*. Este conjunto de políticas permite poner el foco de atención del sistema aprendiz en el tratamiento de aquellos ejemplares no etiquetados que encuentre confusos en base a un cálculo de la distribución de la clasificación de las diversas instancias de datos no etiquetados en el mapa de las posibles clasificaciones que se les pueda imputar. Lo que procura es medir la impureza en el aprendizaje automático. El subtipo que aplicaremos se llama *Entropy Sampling Strategy* [8].

Siendo x_{H_b} el mejor ejemplar en tanto que más confuso escogido por la estrategia basada en la entropía H , θ el límite de decisión que delimita las diferentes clasificaciones posibles, \argmax_x la función [8] que escoge el mejor ejemplar x para el problema dado, y las posibles clasificaciones imputables a x , y $P(x||y)$ la probabilidad de que x sea clasificado como y , su función de distribución la enunciaremos como sigue:

$$x_{H_b} = \argmax_x H_{\Theta}(Y||x) = \argmax_x - \sum_y P_{\Theta}(y||x) \log P_{\Theta}(y||x) \quad (4.5)$$

En nuestro caso, la clasificación es binaria, como veremos más adelante, y la función que

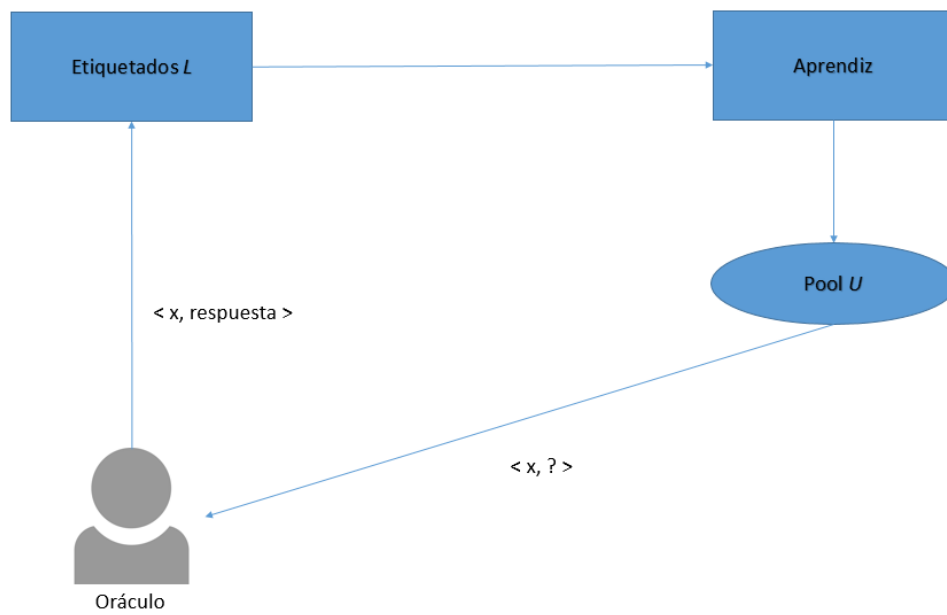


Figura 4.2: Esquema de funcionamiento de AL.

generará la estrategia será monótona, como podremos ver en la Figura 4.3. El algoritmo escogerá el ejemplar que esté más cerca del límite de decisión Θ [8].

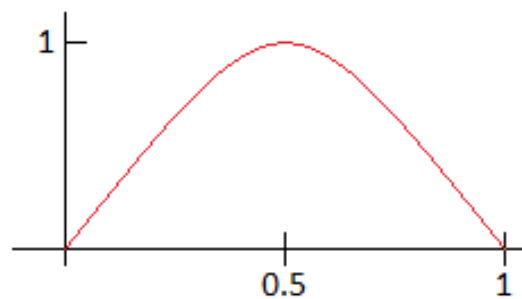


Figura 4.3: Curva de distribución de datos de la estrategia Entropy Sampling.

4.2. Algoritmos supervisados utilizados

En nuestro proyecto usaremos tres clasificadores diferentes. Dos de ellos utilizados siguiendo un esquema de aprendizaje supervisado puro dedicado a la construcción del sistema decisor bio-médico y del sistema predictor ambiental. El tercero, en cambio, será utilizado como base para el

algoritmo de AL que aplicaremos.

A continuación, apuntaremos de forma resumida ciertos desarrollos teóricos sobre los diversos clasificadores supervisados que hemos utilizado.

4.2.1. Red ANFIS

Este clasificador lo hemos utilizado para la construcción del sistema predictor biomédico. Podemos adelantar que el carácter de los datos biomédicos que hemos utilizado para entrenarlo es inestable e impreciso a causa de haber sido extraídos de un entorno práctico real y al concepto impreciso de por sí que encierran. Es necesario pasar esa información difusa previamente por un criterio discriminante que le de una forma discreta. Este criterio lo implementamos con una *red neuronal adaptativa de sistemas de inferencia fuzzy*, también conocida como *ANFIS* por sus siglas en inglés [9].

Lo que hace a la ANFIS interesante es que procura integrar las mejores características de los dos opuestos que lo conforman, tanto de los sistemas fuzzy como de las redes neuronales. De las redes neuronales toma la capacidad adaptativa mediante propagación hacia atrás, y de los sistemas fuzzy toma la representación del conocimiento apriorístico en un conjunto de restricciones (una red topológica) para reducir el espacio de búsqueda de los problemas de optimización (lo que implica que el orden exponencial que referimos está fuertemente podado), aparte de la suavidad de comportamiento que conlleva la interpolación de los controles fuzzy. Dichas ventajas de ambos sistemas se sintetizan y adquieren una nueva significación debido a que la red neuronal tiene un papel doble en el sistema de control fuzzy¹. En primer lugar, implementa un sistema de control de tipo Takagi-Sugeno como una red que se ajusta mediante un método puramente automático y en segundo lugar, la red neuronal implementa un segundo método de control de tipo II que se ajusta mediante un método manual automatizado empleando propagación hacia atrás [9].

En el proceso de aprendizaje, la red neuronal se encarga, mediante los dos métodos referidos, de ajustar dos conjuntos diferentes de parámetros, S_1 y S_2 , de forma automática. El proceso constará de dos subfases diferenciadas, en las cuales uno de los conjuntos estará fijo y el otro se modificará en base al funcionamiento de la red neuronal.

En la primera fase, llamada *paso hacia adelante*, S_1 queda fijo, y S_2 se computa mediante el algoritmo del menor error cuadrático, un problema de regresión lineal.

En la segunda fase, llamada *paso hacia atrás*, S_2 queda fijo y S_1 es computado usando un algoritmo de gradiente descendiente, al estilo de las redes neuronales clásicas en la propagación hacia atrás, que explicamos en el siguiente punto [9].

4.2.2. Perceptrón multicapa

El clasificador dedicado a la construcción del sistema predictor ambiental es un tipo de red neuronal concreto, denominada *perceptrón multicapa*. Esta técnica surgió como alternativa a los acercamientos estadísticos que predominan en los tratamientos de problemas de gran complejidad.

¹<https://es.mathworks.com/help/fuzzy/anfis.html>

Se ajusta a problemas relacionados con las ciencias atmosféricas, y el tratamiento de variables ambientales se encuentra dentro de dicha categoría de problemas [10]. Como veremos más adelante, la disposición y direccionalidad del flujo de información dentro de la red neuronal caracterizan al perceptrón como una *red neuronal de avance*.

El perceptrón multicapa se estructura como un sistema de neuronas interconectadas entre sí, o un grafo de nodos². Tal sistema constituiría un modelo que representa un mapeado no lineal entre una instancia de entrada (x_1, \dots, x_n) y otro vector de salida (y_1, \dots, y_n) . A cada una de las neuronas que constituyen el perceptrón le llegan conexiones de entrada de otras, las cuales tienen un peso asociado (w_1, \dots, w_n) , y son combinadas mediante una suma:

$$z = \sum_{i=1}^n x_i w_i \quad (4.6)$$

Dicha combinación z es computada mediante una función de activación $f(z)$ que genera el valor de salida (y_1, \dots, y_n) . El valor calculado por la función de activación puede ser una nueva entrada de alguna neurona posterior de la red neuronal, o la salida final de uno de los elementos del vector de salida. Estas neuronas básicas se pueden constituir en varias capas tanto parciales como totales interconectadas entre sí que añaden mayor o menor precisión al aprendizaje. La superposición de varias de estas capas es lo que permite la posibilidad de modelar una hipótesis.

El entrenamiento de un perceptrón multicapa, llamado de *propagación hacia atrás*³, se basa en la modificación de los valores de los pesos asociados a las entradas de cada perceptrón, lo que permite calibrar la precisión del modelo generado. Tal calibración se realizaría calculando el error generado por el modelo propuesto, resultado de restar la salida deseada (aprendizaje supervisado) con la señal generada como salida por el perceptrón. El error calculado es propagado hacia atrás a través de la red. Mediante los diversos errores generados y propagados en cada ciclo de calibración, se calibran los pesos de cada neurona. Este procedimiento permite la autoorganización del sistema neuronal.

A nivel general, el algoritmo procura la calibración de los diversos pesos que haya dado el menor error. El primer ciclo tendría valores aleatorios como pesos, y en cada iteración para cada instancia de la masa de datos de entrenamiento el modelo tratará de tender hacia su mejor solución. A este modo de proceder se le denomina *pendiente de gradiente*.

4.2.3. C4.5

Este clasificador ha sido empleado como algoritmo base para la construcción de las diversas hipótesis generadas por el algoritmo AL, quien construye el sistema decisor experto final.

Este algoritmo, llamado C4.5, se trata de una mejora frente a sus dos antepasados, CLS e ID3 [11]. Si bien se puede basar en dos modos de funcionamiento distintos, examinaremos aquel que expresa el problema de aprendizaje mediante un árbol de decisión.

Tenemos que, siendo I un conjunto de instancias, el algoritmo generará un árbol inicial usando un algoritmo de divide y vencerás en función del siguiente criterio. Si todas las instancias en I se

²<http://pendientedemigracion.ucm.es/info/pslogica/redesdoc.pdf>

³https://es.wikipedia.org/wiki/Propagaci%C3%B3n_hacia_atr%C3%A1s

encuentran en la misma clase o si I es pequeño, resulta que hemos encontrado una hoja con la clase más frecuente en S . En otro caso, el algoritmo dividirá el conjunto en varios subconjuntos en función del resultado de un test heurístico realizado sobre el conjunto I a partir de un atributo de que se encuentre en todas las instancias de I . Dicha división del conjunto I dará como producto una serie de subconjuntos I_1, \dots, I_n donde las instancias se encontrarán agrupadas en función del resultado generado por el test. Sobre todos los conjuntos producto de la división se aplicarán de forma iterativa los dos criterios comentados de manera sucesiva [11].

Existen dos heurísticas posibles aplicables como criterio para la realización de los test. La más básica es la ganancia de información, que procura la minimización de la entropía total de los subconjuntos divididos (los I_i). Cuenta con la limitación de que su despliegue está fuertemente sesgado en caso de haber numerosos resultados posibles. Una segunda, más compleja, es el ratio de ganancia por defecto, que divide la ganancia de información con el número de resultados posibles del test [11].

Para evitar que los costes del algoritmo se disparen, pues se ejecuta como un recorrido en profundidad de un árbol, éste cuenta con un procedimiento de poda. Dicho algoritmo se funda en una estimación pesimista de la tasa de error asociada a un conjunto con N clases, siendo E el número de instancias que no pertenecen a la clase mayoritaria. La cota superior pesimista será E/N , y todo error que supere esta cota será podado [11].

Capítulo 5

Metodología

En este capítulo pretendemos mostrar y desarrollar el modelo planteado para nuestro proyecto. Este modelo será la base sobre la que se desarrollará nuestro experimento, buscando cumplir los objetivos que este Trabajo de Fin de Grado se ha planteado. Es decir, demostrar que es posible enriquecer y mejorar la respuesta de un sistema de detección de migrañas fuertemente basado en información hemodinámica mediante información periférica a este sistema, que incluye: variables meteorológicas, pródromos y datos sobre actividad.

En los sucesivos apartados iremos describiendo aquellos módulos que generarán la información que el sistema AL precisa, dejando su caracterización específica para un capítulo posterior, así como el SSD que el propio AL articula.

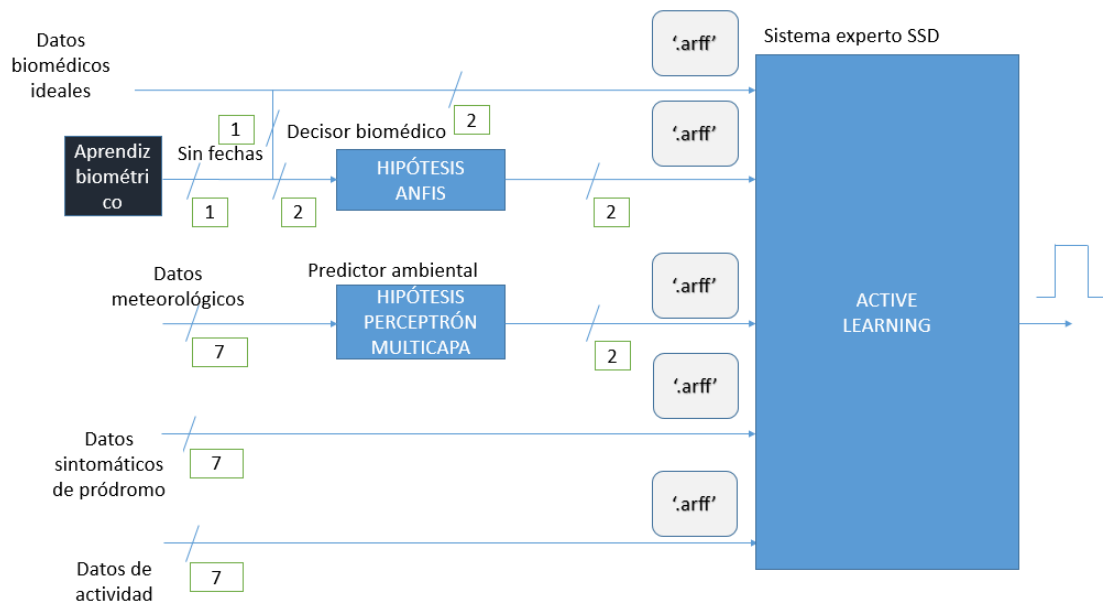


Figura 5.1: Distribución de los módulos del sistema experto SSD.

Como podemos ver en la Figura 5.1, se tratan de dos módulos de aprendizaje supervisado separados del sistema AL pero en interacción que, como dijimos en el Capítulo 4, generan tanto

la alarma biomédica como la alarma meteorológica, y un vector de información proveniente de una base de datos que nos da información de la actividad previa al dolor, más concretamente, del carácter de los síntomas que ha sufrido el paciente y de los datos sobre actividad. Además, a modo de caja negra, se ha agregado un tercer módulo que generará los datos biomédicos difusos que han servido de base para el aprendizaje de la alarma biomédica. El vector de información, debido a su carácter de mera información adicional, será visto en el Capítulo 6.

5.1. Sistema de predicción de migrañas basada en datos biomédicos

Este módulo, si bien no se encuentra activamente en nuestro proyecto, sí ha generado información crucial para el mismo. Lo supondremos dentro del proyecto como una caja negra. En nuestro modelo, como refleja la Figura 5.1, se encuentra detrás del bloque decisor biomédico.

Se trata de un módulo que entrena diversos modelos para cada variable hemodinámica obtenida del paciente con la intensidad de dolor (ver Figura 5.2). En concreto, TEMP, ED, HR y SP02 [1]. Utilizan como base algoritmos de espacio de estados de tipo supervisado, siendo indistintamente o N4SID [12] o un algoritmo basado en Gramatical Evolution¹ desarrollado por el equipo de trabajo GreenDISC del grupo de investigación ArTeCS de La Universidad Complutense de Madrid [13].

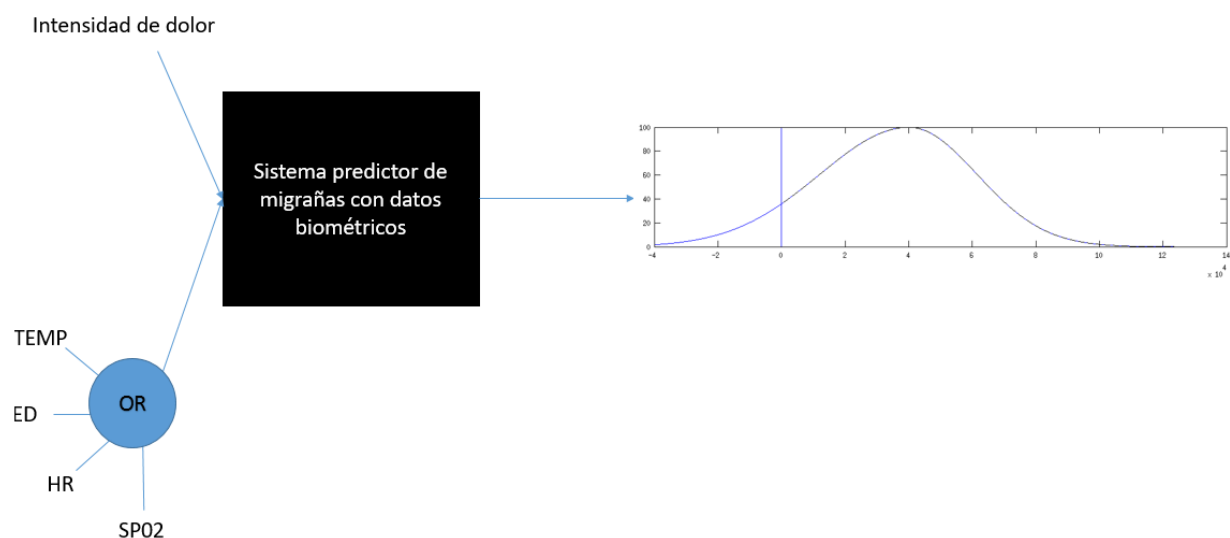


Figura 5.2: Entrada y salida de la caja negra.

Este módulo nos genera predicciones de migrañas que idealmente tienen forma de dos curvas semigaussianas con μ y σ distintos, tal como se muestra en la Figura 5.2. Hay que apuntar que las campanas gaussianas se fundan en una hipótesis, la cual se encuentra explicada en este artículo [3]. Tales curvas serán utilizadas por nuestro módulo productor de la alarma biomédica, como veremos a continuación.

¹https://en.wikipedia.org/wiki/Grammatical_evolution

5.2. Sistema decisor biomédico

El siguiente módulo se trata de un sistema decisor que determina cuándo se produce o no una migraña, produciendo así la alarma biomédica como salida. Como podemos ver en la Figura 5.1, se sitúa entre la caja negra y el sistema decisor experto.

La información de entrada son las diversas curvas generadas por la caja negra. Su cometido es binarizar la información que le llega, prescindiendo del umbral que se ha usado en otros casos, como podremos ver en la siguiente figura (ver Figura 5.3).

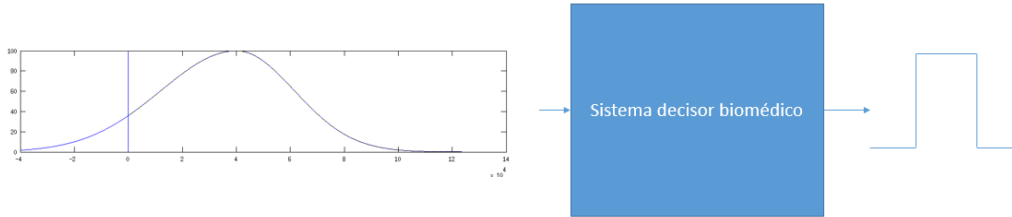


Figura 5.3: Entrada y salida del sistema decisor.

Para ello, hemos confiado en el algoritmo de aprendizaje ANFIS. En concreto, hemos utilizado su implementación de MATLAB. Por restricciones técnicas, ejecutamos ANFIS como un módulo aparte y aislado del AL.

Los datos que trata el sistema decisor tienen una naturaleza de por sí imprecisa. Los límites de comienzo del dolor no son claros y son variables en cada migraña por su forma curva, aspectos que hacen a este problema ideal para ser tratado con la lógica difusa que implementa ANFIS. De lo continuo nos dará una respuesta discretizada.

Para el entrenamiento y test hemos planteado mediante ANFIS un algoritmo propio, desarrollado en código MATLAB, del que explicaremos su idea general.

Sea P el conjunto de los pacientes y $D(x)$ la señal de dolor del paciente x , asumimos como hipótesis que, $\forall p \in P : D(p)$ tienen una forma similar. Dolor/no dolor o curva gaussiana. Al introducir los diversos $D(p)$ en el ANFIS para su entrenamiento y testeo, lo haremos prescindiendo de sus etiquetas temporales, considerando los tiempos en que se producen los instantes de dolor y no dolor como ideales. Centraremos el aprendizaje del sistema en la forma de la curva de intensidad y no tanto en la temporalidad particular en que ésta se desarrolla. Entrenaremos modelos $\forall q \in P : h_{anf_{is}_q}$ de los que se elegirán dos para emplearlos de forma combinada.

Para aligerar la carga del entrenamiento en aras de un rendimiento más asumible y un proceso de aprendizaje más transparente, realizaremos una validación cruzada siguiendo un modelo de *dejar uno fuera*. Dividiremos a los pacientes en $X \subset P$: conjunto de entrenamiento e $Y \subset P$: conjunto de test, cumpliéndose que $X \cap Y = \emptyset$. $\forall x \in X$ construiremos una $h_{anf_{is}_x}$ mediante un conjunto de entrenamiento $X - x$ y se utilizará a x como conjunto sobre el que ejecutar la validación de la hipótesis, la cual arrojará un determinado $fit_{h_{anf_{is}_x}}$. Con ello se construirán $\|X\|$ hipótesis en total, con el mismo número de fits, una para cada una. Para el cálculo del fit discre-

tizaremos la salida que arrojen los diversos modelos ANFIS contruidos mediante un umbral del 35 % (con un 50 % de fiabilidad) [3], el cual nos permitirá calcular más fácilmente las derivadas de subida. $fit_{hanfisx}$ se regiría, matemáticamente expresado, por la siguiente fórmula, siendo t_{fuzzy_i} el instante de tiempo de la curva decisión del ANFIS en que la derivada de la curva sea 1 y t_{real_i} lo mismo para el caso de la señal de dolor real en caso de que hayan k migrañas detectadas:

$$fit_{hanfisx} = \frac{1}{k} \sum_{i=1}^k t_{fuzzy_i} - t_{real_i} \quad (5.1)$$

En concreto, este fit calculado corresponde al error medio temporal de inicio del dolor de migraña en caso de que la señal arrojada por el ANFIS con respecto a la real se adelante o atrase. El siguiente ejemplo gráfico de la Figura 5.4 nos permitirá hacernos una idea de a qué nos referimos.

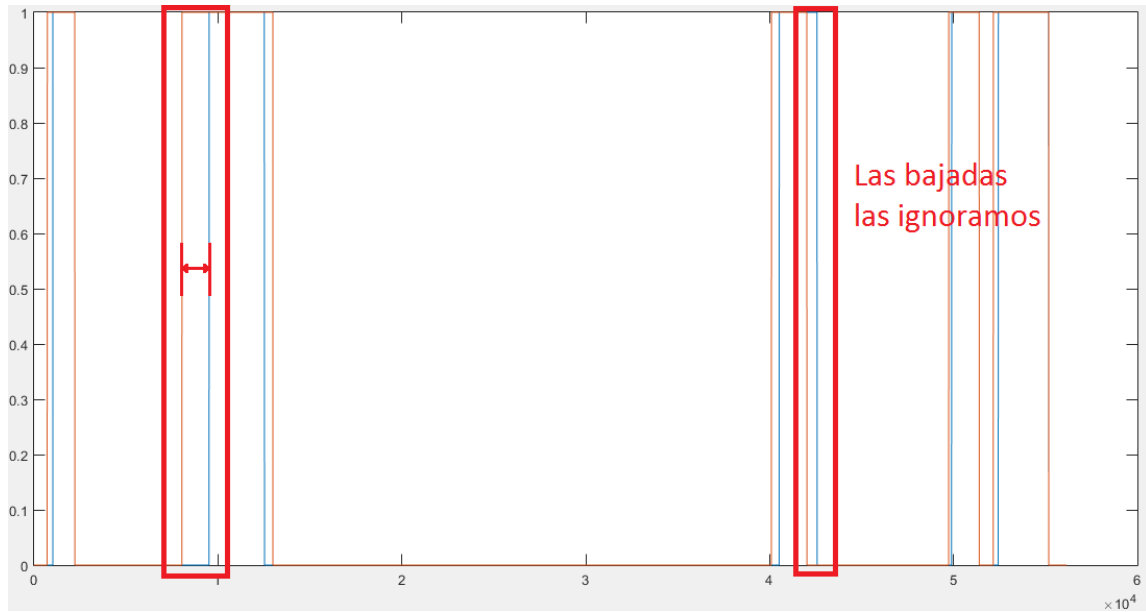


Figura 5.4: Adelantamiento y retraso de la respuesta del sistema decisor biomédico para el cálculo del fit. En la figura, la gráfica azul es la respuesta real, mientras que la respuesta dada por nuestro sistema es la naranja

Después, elegimos aleatoriamente dos de los modelos calculados. El primer modelo escogido se encontrará entre aquellos que hayan sido testeados con los datos de los pacientes del año 2014, mientras que el segundo modelo escogido se encontrará entre aquellos que hayan sido testeados por los pacientes del año 2017. Este criterio se debe a la amplia diversidad cualitativa de los datos entre ambos períodos, que discutiremos en detalle en la Sección 6.1. Posteriormente, ambos modelos atenuarán su desviación mediante el entrenamiento con X y serán testeados sobre Y , generando como producto sobre sus conjuntos correspondientes de pacientes la salida del sistema.

Además, junto con los fits de los conjuntos de test, incorporaremos otra métrica, la matriz de confusión, que nos permitirá hacernos una idea de los falsos positivos o negativos generados,

siguiendo este criterio:

- Positivo verdadero: el sistema detectó bien.
- Positivo falso: hubo una invención del sistema.
- Negativo falso: hubo una omisión del sistema.
- Negativo verdadero: el sistema está en reposo.

El criterio de selección de modelo que planteamos en primera instancia nos fracasó, pues se basó en el cálculo de la desviación típica de los diversos fits y su filtro en función de un ϵ cercano a cero. Sin embargo, como veremos en la Sección 7.2, entre los diversos modelos nos encontramos con gran disparidad de resultados, por lo que optamos por rechazar este criterio.

Para terminar con el módulo, apuntamos que en primer y segundo lugar el clasificador ANFIS no iba a ser la primera opción de implementación fuzzy que habíamos considerado y que, en su lugar, iba a ser utilizada la librería para Java *JFuzzyLogic* en primer lugar, construyendo nuestro propio sistema de aprendizaje y en segundo lugar *FURIA*, una clase integrable en WEKA que contenía el aprendizaje fuzzy ya implementado. En el primer caso, consideramos que debíamos buscar una alternativa que no requiriese ningún tipo de conocimiento, más allá de los datos a entrenar. Además, concluimos que era una mejor opción que la librería estuviera 100 % integrada en WEKA. *FURIA* fue una respuesta, una librería para WEKA que era capaz de inferir por sí misma el sistema fuzzy. Sin embargo, descubrimos que era una solución muy lenta y con poco feedback para el usuario durante el proceso de aprendizaje. ANFIS no mejora esta latencia abultada, pero la construcción que hemos planteado y que hemos explicado sí que aligera el tiempo de respuesta e incorpora más feedback del desarrollo del proceso con métricas a nuestra medida.

5.3. Sistema de predicción bajo variables ambientales

El siguiente módulo se trata de un sistema de predicción cuyo cometido es buscar asociaciones informativas de tipo ambiental con respecto a la señal de dolor de los pacientes (ver Figura 5.5). Como podemos ver en la Figura 5.1, se encuentra en segundo lugar.

Dado que existen controversias en el ámbito de la comunidad científica sobre cuáles son las variables ambientales concretas que funcionan como disparador de la migraña, nosotros dejamos que el módulo construya la hipótesis pertinente en base a lo que tenemos, confiando en que nos dé una buena respuesta.

Para la construcción de tal hipótesis hemos confiado en un algoritmo de aprendizaje basado en el perceptrón multicapa, en concreto, en una implementación interna a la librería WEKA. El módulo se ejecutará en un archivo Java independiente, aislado del sistema experto, como en el caso anterior, y su empleo no se distinguirá en cuanto a ejecución del uso habitual de los algoritmos de WEKA.

Para validar el proceso de entrenamiento empleamos el método de validación cruzada de 10-particiones.

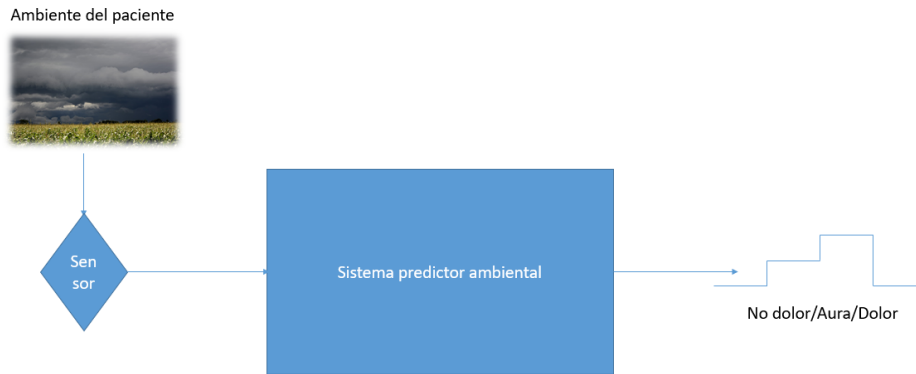


Figura 5.5: Entrada y salida del predictor ambiental.

Relacionadas con este proceso, empleamos dos métricas para determinar la corrección de su uso: TPR, o *tasa de verdaderos positivos*, también llamada *exhaustividad*², la cual se define en función de la siguiente fórmula:

$$TPR = \frac{vp}{vp + fp} \quad (5.2)$$

Y TPV, o *valor de predicción positiva*, también llamada *precisión*, la cual se define según la siguiente fórmula:

$$TPV = \frac{vp}{vp + fn} \quad (5.3)$$

Siendo vp el número de verdaderos positivos, fp el de falsos positivos y fn el de falsos negativos.

Sin embargo, como métrica, no nos centramos en usar el error de la validación cruzada, el cual es una media de los TPR de cada partición, al igual que ocurre con sus TPV. El valor del error de la validación cruzada resulta insuficiente para comprender cuál es el error real que el sistema de aprendizaje arroja en su proceso de entrenamiento. La distribución de las distintas clasificaciones en los datos escogidos como información de entrenamiento provoca un sesgo en el error. No se tienen en cuenta los aciertos y los errores de las diversas clasificaciones como tal, aislados de su frecuencia de aparición, si no que se ven en su conjunto de un modo muy ciego. Si, por ejemplo, de una clasificación binaria de 0 y 1, nos encontramos con que hay muchos ejemplares clasificados con 0 y otros ejemplares de menor frecuencia de aparición clasificados con 1, tendremos que el error de validación cruzada estará fuertemente sesgado a la clasificación 0, y no nos dirá demasiado del error de clasificación que haya podido sufrir el entrenamiento para la clasificación 1.

Por suerte, existe otra métrica que nos permite sortear las limitaciones que esta métrica anterior implica llamada *F-Valor*. Se trata de una medida para determinar la precisión que tiene un test. Se sirve del cálculo de un valor único ponderado de la precisión y de la exhaustividad. Permite aislar

²https://en.wikipedia.org/wiki/Precision_and_recall

la efectividad del algoritmo según la calidad de la clasificación para cada una de las clasificaciones posibles, evitando la repercusión de la frecuencia de aparición concreta de las clases. Sean $\beta \in R$, x la precisión e y la exhaustividad, la función que lo rige es la siguiente:

$$F_{\beta} = 1 + \beta^2 \frac{xy}{(\beta^2 x) + y} \quad (5.4)$$

Dependiendo del valor de β , uno de los dos atributos tendrá la principalidad en cuanto a influencia en el resultado. Si $\beta = 1$, ambos se encontrarán en equilibrio, si $\beta \geq 1$ entonces la exhaustividad será el principal, mientras que si $\beta \leq 1$ será la precisión el aspecto determinante.

5.4. Sistema experto SSD basado en Active Learning

El último módulo que trataremos será el sistema experto principal, que recibe toda la información del resto de módulos y trata de construir el modelo de predicción definitivo, persiguiendo mejorar la respuesta de sus predecesores basada en umbral. En la Figura 5.1 es el bloque que preside el lado derecho.

Se valdrá de un algoritmo AL que modela la alarma definitiva, y como algoritmo base para la construcción de las hipótesis el C4.5. La implementación utilizada para el caso del AL será la librería JCLAL, y para el algoritmo C4.5 escogeremos su implementación en WEKA, llamada J48, que como método de validación emplea el algoritmo *Hold Out*. El módulo se encuentra en un archivo Java independiente, cuya ejecución está precedida por un algoritmo de enlazamiento de datos para la construcción del archivo definitivo a usar en el sistema experto.

La librería JCLAL se basa fundamentalmente en los principios vistos en el Capítulo 4 seguidos punto por punto. Es una librería basada en JCLEC, desarrollada bajo licencia GPL [14] por el grupo KDIS research group. Asegura que ofrece un buen conjunto de estrategias AL pertenecientes al estado del arte para el tratamiento de datos de una etiqueta³. Esta librería nos da un entorno integrado en WEKA e interactivo, que nos permite introducir nueva información a la par que mejorar el proceso de aprendizaje.

Nos centraremos en el entorno que JCLAL nos ofrece para el desarrollo de nuestro experimento y que nos permite interactuar activamente con él. Consta de un sistema de introducción de clases a través de una clave numérica mediante línea de comandos, como podemos ver en la Figura 5.6), que nos irá preguntando las diversas instancias a las que clasificar manualmente. Cuenta con una ventana aparte que nos mostrará las gráficas de las diversas estadísticas que arroja el sistema para las diversas iteraciones en las que se aplica, dándonos así una idea del proceso global(ver Figura 5.7) de entrenamiento. Sin embargo estas estadísticas están limitadas, por lo que hemos profundizado en las capacidades de la librería para poder conseguir el TPR, el TPV y el F-Valor para las evaluaciones de cada iteración.

Mediante este entorno pretendemos completar unos datos que no tenemos. En concreto, un ruido que hemos introducido en el tiempo que media entre el inicio del prodrómico y el inicio de dolor que incluye al abarcado por el aura. En dicho tiempo faltará la clasificación, permitiéndonos

³<https://sourceforge.net/p/jclal/git/ci/d118a344e8cb0bde28e23345083cbeb7274c8c69/tree/jclal-documentation.pdf>

```

Human oracle.
What is the class of this instance?
Instance: 201703111511,0.75246,0,0,0,0,0,0,0,0,0,0,0,0,0,0,NP

Index: Class name
-----
0: NP
1: P

Type the index of the class or type -1 if you want skip this instance
index >>

```

Figura 5.6: Entorno de interacción máquina-usuario de JCLAL.

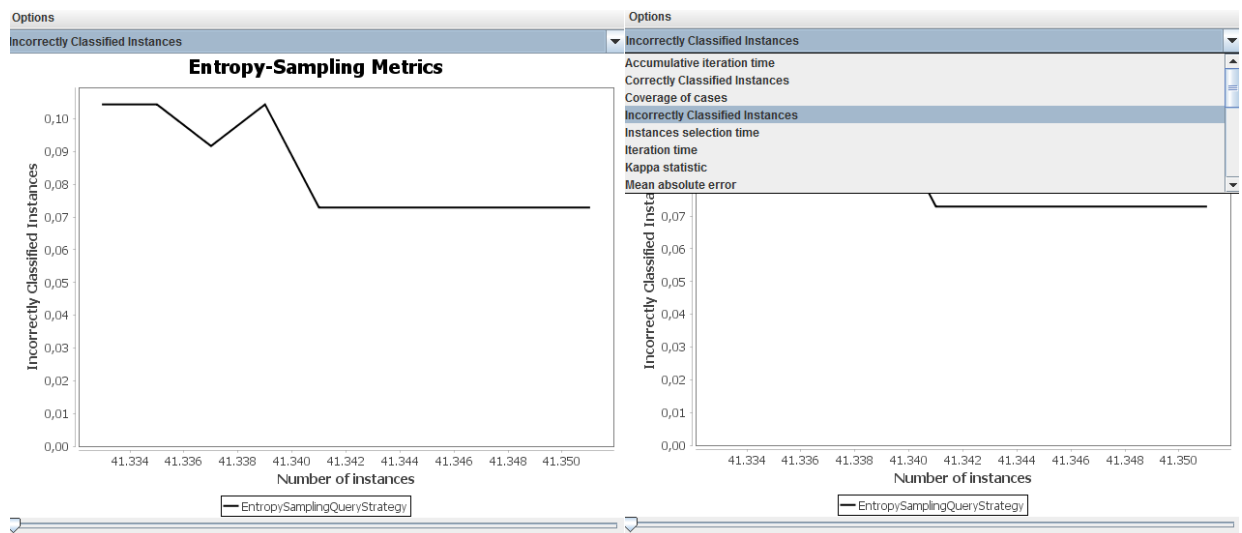


Figura 5.7: Ventana de presentación de gráficas JCLAL.

incorporar cierta información. También buscamos que el propio sistema experto consiga de manera independiente rellenar esos huecos por sí mismo con alarmas de dolor o no dolor en función de lo que considere, y de cómo comprenda en concreto lo que son el aura y el prodrómico. Por tanto, aquí caminamos sobre una hipótesis y no conocemos los posibles resultados que puedan darse.

Capítulo 6

Fase experimental

En el siguiente capítulo cubrimos los aspectos que rodean al experimento: los datos utilizados, de qué adolecen y cómo han sido tratados, aparte de los procedimientos experimentales seguidos en el experimento para que otro investigador pueda replicar sus condiciones y, en mayor o menor medida, sus resultados.

En primer lugar, cubrimos los orígenes y el tratamiento de los datos que introducimos y que obtenemos de los dos módulos de aprendizaje supervisado para pasar a detallar el vector de información de prodrómicos y datos de actividad. Así, estudiamos en profundidad la tupla final que se inserta en el sistema experto.

En segundo lugar, examinamos el procedimiento experimental que hemos seguido para la obtención de los resultados, que consignamos detalladamente en el Capítulo 7.

6.1. Obtención y tratamiento de datos

6.1.1. Datos biomédicos

La información de entrada que introducimos al módulo que tratamos en la Sección 5.2 está construida como un conjunto de instancias que albergarán los siguientes datos de entrenamiento y test:

- La curva gaussiana calculada por la caja negra referida en Sección 5.1, cuya forma podemos ver en la Figura 6.1.
- La información de dolor que han reportado los diversos pacientes, sin las auras que han sufrido.

La salida generada tiene una forma de matriz tiempo-dolor-paciente que será introducida al sistema experto directamente. El dolor se modela siguiendo una forma continua, como podemos ver en la Figura 6.2. Tanto la información de la curva de intensidad de dolor como la información de dolor tienen asociadas una etiqueta temporal distribuida en minutos que no se utilizará en el entrenamiento. Consideraremos el tiempo como ideal.

En el experimento hemos dispuesto de dos conjuntos de datos de dos años diferentes. El primero, de pacientes del año 2014 y el segundo de pacientes del año 2017. El primer conjunto consta

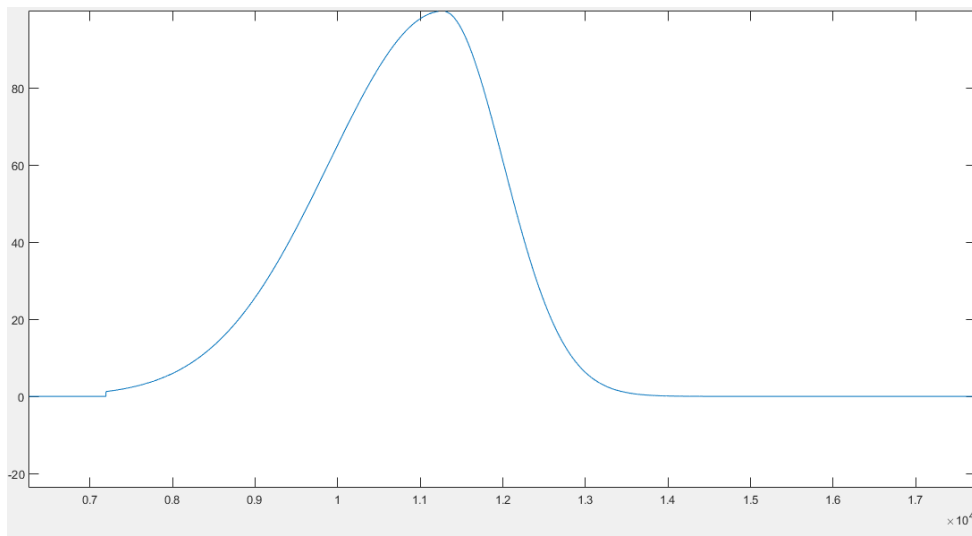


Figura 6.1: Curvas gaussianas generadas por la caja negra.

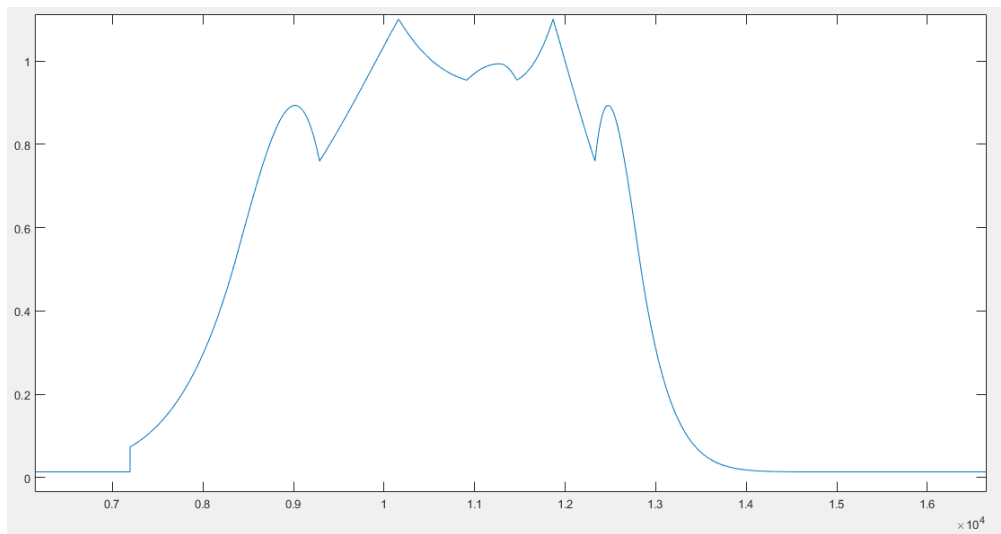


Figura 6.2: Pulso de dolor generado por el sistema experto.

de 12 pacientes, mientras que el segundo consta de 13 pacientes. Los datos biomédicos calculados por la caja negra se han conseguido mediante dos sensores distintos para cada conjunto. En el primer caso se han empleado unos sensores biométricos *BioPlux*, desarrollados por la compañía Plux¹. En el segundo, se ha utilizado un sistema de sensores ad-hoc con capacidad de computación distribuida desarrollado por el equipo de trabajo *GreenDISC* del grupo de investigación ArTeCS

¹<http://www.plux.info/index.php/en/>

de La Universidad Complutense de Madrid.

La información ha sido distribuida siguiendo su temporalidad real, procurando que la forma de la curva sea lo más veraz posible. Para separar los extremos entre paciente y paciente hemos incorporado una separación entre pacientes de lo que reste del primer o último día de la migraña.

La información binaria de dolor para el entrenamiento de la alarma final y la alarma aplicada en nuestro sistema decisor ANFIS ha sido proporcionada por los pacientes de dos modos diferentes. El conjunto de pacientes del año 2014 proporcionó dicha información mediante un formulario médico, en un ambiente de monitorización ambulatoria. Por otro lado, los pacientes del año 2017 proporcionaron la información mediante una app móvil llamada BrainGuardApp, desarrollada por el equipo de trabajo *GreenDISC* del grupo de investigación ArTeCS de La Universidad Complutense de Madrid. En ella los pacientes indicaron los diversos eventos que han sucedido en el proceso de la migraña.

Esta información la hemos distribuido en la línea temporal, siguiendo unos criterios similares a los de las curvas gaussianas, con distintos valores minuto a minuto en función del evento:

- Valor P (1): si hay evento de dolor.
- Valor NP (0): si no hay evento de dolor, ni de pródromo o aura.
- Valor desconocido (?): si hay evento de pródromo o aura, siendo desconocido debido a que no sabemos cómo podría comportarse el sistema experto final en estos casos.

En la alarma biomédica con la que entrenamos a nuestro sistema decisor ANFIS, a diferencia de la alarma final entrenada, no empleamos los datos desconocidos, sustituyéndolos por ceros, lo que implicaría que analizamos si hay dolor o no dolor.

Debido a las limitaciones causadas por la extracción de datos de pacientes reales inmersos en las más diversas circunstancias, muchas de las curvas no se acercan a la forma ideal debido a la imperfección de la información de la intensidad de dolor. De los pacientes antiguos del 2014 no hay datos de variación de intensidad del dolor, y, en algunos casos, de pacientes modernos de este año, o los datos de intensidad de dolor proporcionados por los pacientes son muy defectuosos, o se han perdido o son directamente inexistentes. Lo que hemos hecho en caso de no haber información para la curva ha sido llevar a cabo el procedimiento que planeamos originalmente, introducir el pulso de dolor tal cual (con valores de 0 y 100, como podemos ver en la Figura 6.3) con un ruido en el inicio y en el final del dolor, cuya duración sea $\|t_r\| \leq 10$ minutos en cada extremo. Sospechamos que el carácter simple de esta información ha hecho que sea trivial para la construcción de la hipótesis. Aun así hemos corrido ese riesgo en aras de que la hipótesis construida refleje también estos casos. En algunos casos aislados, debido al carácter defectuoso de algunas migrañas, éstas han sido eliminadas.

6.1.2. Datos ambientales

La información de entrada que hemos introducido en el módulo referido en la Sección 5.3 consta de un conjunto de instancias que contienen como información una tupla de los datos am-

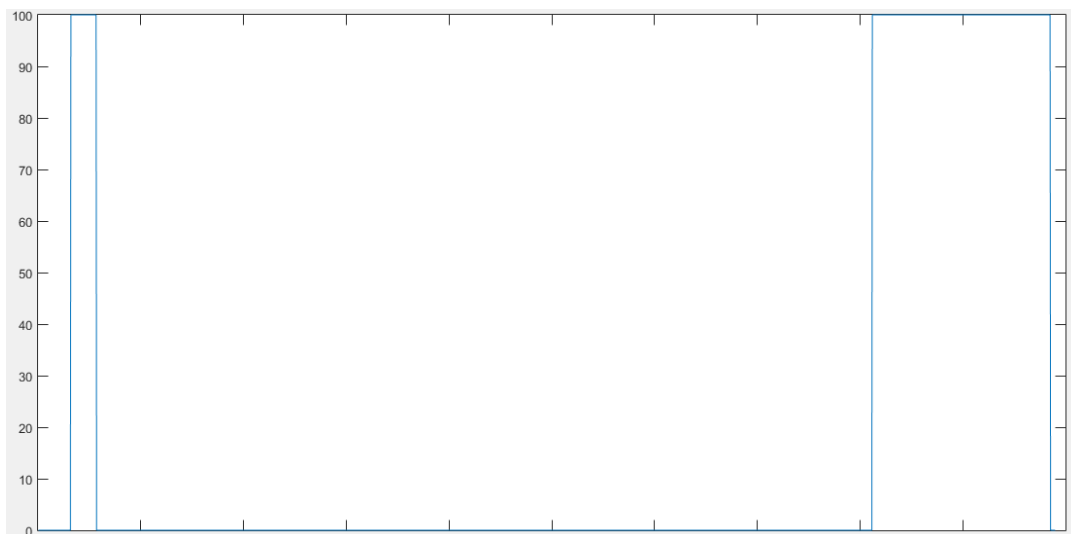


Figura 6.3: Curvas de tipo 0-100.

bientales en los cuales se han encontrado inmersos el conjunto de los pacientes, con los siguientes datos:

- La temperatura ambiente, en grados centígrados.
- Las precipitaciones, en milímetros.
- La presión atmosférica, en hectopascales.
- El diferencial de presión máxima, máxima pendiente de la curva generada por los datos de presión atmosférica de la última hora.
- La humedad, en porcentaje.
- La información del dolor, referida anteriormente, como alarma a entrenar para su modelado.

El módulo produce como salida una matriz de datos tiempo-dolor-paciente, que hemos introducido directamente al sistema experto. La tupla de información ambiental tiene asociada una etiqueta temporal distribuida en horas. La información temporal es importante en el proceso de aprendizaje del predictor y la hemos incorporado en los conjuntos de entrenamiento y test. El entrenamiento sigue un acercamiento de grano grueso debido a que lo entrenamos para cada hora. En el enlazamiento de datos final para el sistema experto, cada hora se asigna al minuto correspondiente para el entrenamiento de la alarma final mediante un proceso de interpolación.

En el experimento hemos dispuesto de dos conjuntos de datos de dos años diferentes, del 2014 y del 2017. Se han empleado dos sensores de recogida de información ambiental diferentes para su obtención. Para los pacientes del conjunto de datos del 2014 se han empleado los datos recogidos por los sensores de *AEMET*², en concreto, los localizados en el Parque del Retiro, el más

²<http://www.aemet.es/es/portada>

cercano al hospital Gregorio Marañón. En el caso de los pacientes del conjunto de datos del 2017, se ha empleado una API que emplea sensores libres desarrollada por el grupo *Weather Underground*³. Esta API nos ha permitido generar la información requerida. La localización empleada se encuentra cerca del Hospital Princesa de Madrid.

Si bien la alarma a entrenar está construida con los criterios de la alarma biomédica, los diversos valores que puede tomar son diferentes, considerando el siguiente criterio:

- Valor P (2): si hay evento de dolor.
- Valor A (1): si hay evento de aura.
- Valor NP (0): si no hay evento de dolor y no de aura.

Debido a la distribución desigual de las clasificaciones de las diversas instancias no pudimos neutralizar el sesgo de información sobre los datos de entrenamiento y test. Como indicamos en el Capítulo 5, utilizamos la métrica F-Valor, evitando esta situación en su repercusión en las métricas.

En el conjunto de los datos meteorológicos se ha producido una pérdida de información en algunos tramos a causa de las limitaciones de los sistemas de recogida de información ambiental consultados. Como metodología de reconstrucción de datos, para los pacientes del año 2014, hemos rellenado con las medias para cada hora de los datos disponibles, evitando el impacto sobre el sistema. Para los pacientes del año 2017, la pérdida es más acusada y sistemática entre las 10 de la noche y las 7 de la mañana. Hemos tomado los datos de la última hora del día y de la primera hora del día y los hemos extendido hacia las 12 de la noche para cada período. Se produce un cambio brusco a las 12 de la noche entre ambos días, pero consideramos que esta apreciación es realista. Se suma que la información de los pacientes del 2017 es más simple que la información de los pacientes del 2014, no existiendo datos de precipitaciones y con datos de presión sin decimales.

6.1.3. Datos prodrómicos y de actividad

El siguiente conjunto de datos que hemos tratado contiene la información detallada sobre los síntomas del pródromo y la información de los diversos disparadores de actividad asociados a las migrañas. Esta información se compondrá de dos tuplas de datos, una con los diversos síntomas que el paciente sufre en el período prodrómico, a saber:

- Síntomas relacionados con el humor.
- Síntomas relacionados con el apetito.
- Síntomas relacionados con el sueño.
- Síntomas relacionados con aspectos cognitivos.
- Síntomas relacionados con aspectos térmicos del paciente.

³<https://www.wunderground.com/>

- Síntomas de tipo gastrourinario.
- Otros síntomas que no entren en estas categorías.

Y otra, con los diversos datos de los disparadores de actividad que han podido causar la migraña, a saber:

- Causados por algo que se ha comido.
- Causados por algo que se ha bebido.
- Causados por sometimiento a estrés del paciente.
- Causados por eventos ligados al sueño.
- Causados por eventos ambientales
- Causados por actividad deportiva.
- Otras causas que no entren en estas categorías.

Además, cada instancia tiene asociada una etiqueta temporal distribuida en minutos para poder asociarlas a las producidas por los otros dos módulos.

Buscamos enriquecer al sistema experto mediante la incorporación de información previa y causante de la migraña. Ambas tuplas no son base para la construcción de ningún modelo.

De comenzar a producirse determinados patrones de pródromo sufridos por el paciente, el sistema experto puede establecer algún tipo de asociación entre estos datos y la migraña que se desarrolle. Se asocia a la migraña como una mirada hacia el futuro, fundamentada en el análisis del pródromo.

Si ciertos disparadores de actividad tienen presencia en los hábitos del paciente, es posible que el sistema experto establezca un patrón entre su causalidad y el estallido de la migraña.

La fuente de los datos para ambos conjuntos de pacientes, los del 2014 y los del 2017, son las mismas que las fuentes de los datos biomédicos.

El uso de los síntomas del pródromo sólo puede darse en la fase de entrenamiento y no en tiempo real. Son variables subjetivas, que los pacientes cumplimentan mediante un formulario o una aplicación móvil antes del desencadenamiento de la migraña, en ocasiones no completadas. Si se desencadena la migraña y los pacientes no han indicado ningún síntoma, al indicar que el dolor ha cesado, se le pregunta al paciente cuáles han podido ser los síntomas sufridos y el tiempo en que, según ellos, comenzaron y finalizaron. Los tiempos y su aparición son, a veces, una estimación realizada por el propio paciente después de haber pasado por el episodio de migraña. En otros casos la información del final de los síntomas no existe, para lo que suponemos que la información de final de síntomas es el inicio del episodio de dolor. Esta suposición es buena, pues la mayoría de finalizaciones de síntomas en los diversos pacientes que sí aportan esta información suelen terminar nada más comenzar el episodio de dolor. Sin embargo, la causa de su no existencia se debe a que los síntomas de pródromo se solapan con el inicio de dolor. A veces el carácter del síntoma no

está debidamente cumplimentado por el paciente, aunque podamos encontrar información sobre el inicio o sobre el inicio o final de los síntomas. Lo que hemos hecho es asignar esas periodizaciones a todos los síntomas por igual. Hemos procurado afirmar que, en ocasiones, la predicción de la migraña a futuros no tiene por qué estar asociada a un conjunto limitado de síntomas en caso de carecer de información sobre el carácter de los síntomas. Sin embargo, la incidencia estructural de estos casos ha resultado ser bastante baja y es muy probable que el impacto en la construcción de la hipótesis resultado de la aplicación del sistema experto final haya sido débil. De no existir los datos de síntomas han sido cubiertos por defecto con ceros.

Sobre los datos de actividad, si bien faltan en algunos pacientes por causas similares a las referidas en los datos de síntomas prodrómicos, su integridad es mayor. No hemos realizado tratamientos sobre los datos más allá de los usuales en cuanto a su despliegue en el tiempo, asociando su inicio y final a la duración del dolor de la migraña en que actúa. De no existir los datos de actividad se cubren por defecto con ceros.

6.2. Procedimiento experimental

El despliegue procedimental que hemos desarrollado para la realización del entrenamiento offline se basa en la utilización de varios modelos de predicción modularizados y aislados entre sí debido a las altas latencias de algunos de ellos. Han sido ejecutados independientemente, generando en cada caso diversos archivos que han sido puestos en relación manualmente. El procedimiento seguirá el modelo de la Figura 5.1, paso por paso.

En el caso del modelo ANFIS, hemos referido su algoritmo de entrenamiento en la Sección 5.2, y ha sido ejecutado en MATLAB mediante el fichero *anfisBio.m* que lo encontraremos en la carpeta *anfisBio*. Debe tener en la misma carpeta en la que *anfisBio.m* figura los archivos de los pacientes que vamos a tratar, cuyo nombre sigue una estructura tipo *i-m-numeropaciente.csv*, aceptando sólo a los pacientes que hemos tratado. No se trata de una implementación genérica. Tras el proceso nos ha generado un archivo llamado *biomedic-pain-all-patients.csv*, al que es necesario poner la columna de fechas a mano, las cuales están alojadas en un fichero llamado *dates.csv*. También ha sido necesario incorporarle una cabecera que figura en el archivo *header.txt* en el inicio del csv, cambiando su extensión al guardarlo de .csv a .arff. En una computadora común, el desarrollo del proceso de aprendizaje del ANFIS es muy lento debido a la alta complejidad de sus operaciones. La latencia de respuesta se encuentra en torno a las dos horas.

Para el modelo generado por el perceptrón multicapa hemos dispuesto de un archivo llamado *weather-merged.arff*, y con él se ha realizado un entrenamiento con un conjunto de datos del 70 % con respecto al original, validado dicho entrenamiento mediante validación cruzada. El resto ha sido empleado para realizar el testeo del entrenamiento. El módulo se encuentra implementado dentro del proyecto Java *ES-AL*, el cual contiene en el paquete *Weather* una clase Java llamada *weatherConstruct.java*. Ejecutamos el archivo, con un entorno Java cualquiera, teniendo el archivo de datos de entrenamiento y test en la raíz del proyecto. Tras el entrenamiento y el test, el modelo calculado clasifica la masa de datos completa. Tenemos que, al terminar, nos ha generado un archivo *weather-pain-all-patients.arff* al que no es necesario incorporarle manualmente la

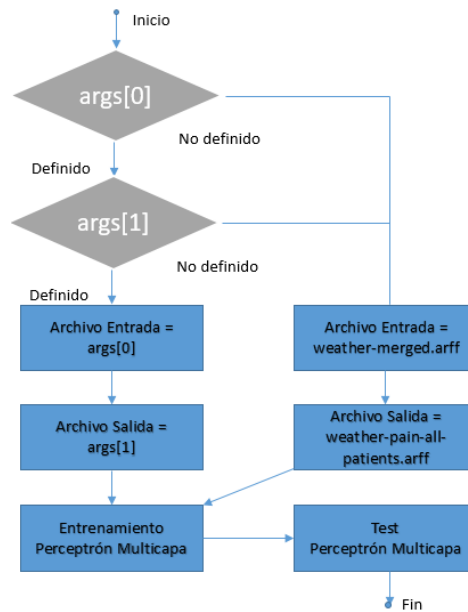


Figura 6.4: Diagrama de flujo del predictor ambiental.

cabecera. En este caso, debido al bajo peso de la información procesada, el predictor tarda menos de un minuto en entrenarse y realizar el test. En aras del estudio profundo de los datos, hemos realizado entrenamientos y test en dos experimentos distintos con los datos aislados por año y así estudiar sus posibles influencias. Como podemos ver en el diagrama de flujo que muestra la Figura 6.4, es posible modificar el archivo fuente de datos introducido al programa metiendo en la entrada del programa el nombre del archivo que deseamos que cargue, además del nombre de un archivo de salida donde se guarden los resultados de la clasificación. En caso de que falte alguno de estos datos se utilizan las rutas por defecto. Esta posibilidad de modificación de los archivos de entrada y de salida nos ha servido para realizar la ejecución de los conjuntos de pacientes por separado, cuyos datos hemos dividido manualmente.

El resto de datos se encuentran en los archivos *symptoms.arff*, donde se encuentra la información de los síntomas prodrómicos, y *activity-t.arff*, donde se encuentra la información referida a la actividad, los cuales no deben ser entrenados en ningún módulo aparte, si no meterse al sistema experto directamente, tal como se muestra en la Figura 5.1.

Todos los archivos remanentes de los procesos de entrenamiento anteriores deben ser llevados al sistema experto final con los nombres que hemos referido:

- biomedic-pain-all-patients.arff
- weather-pain-all-patients.arff
- symptoms-t.arff
- activity-t.arff

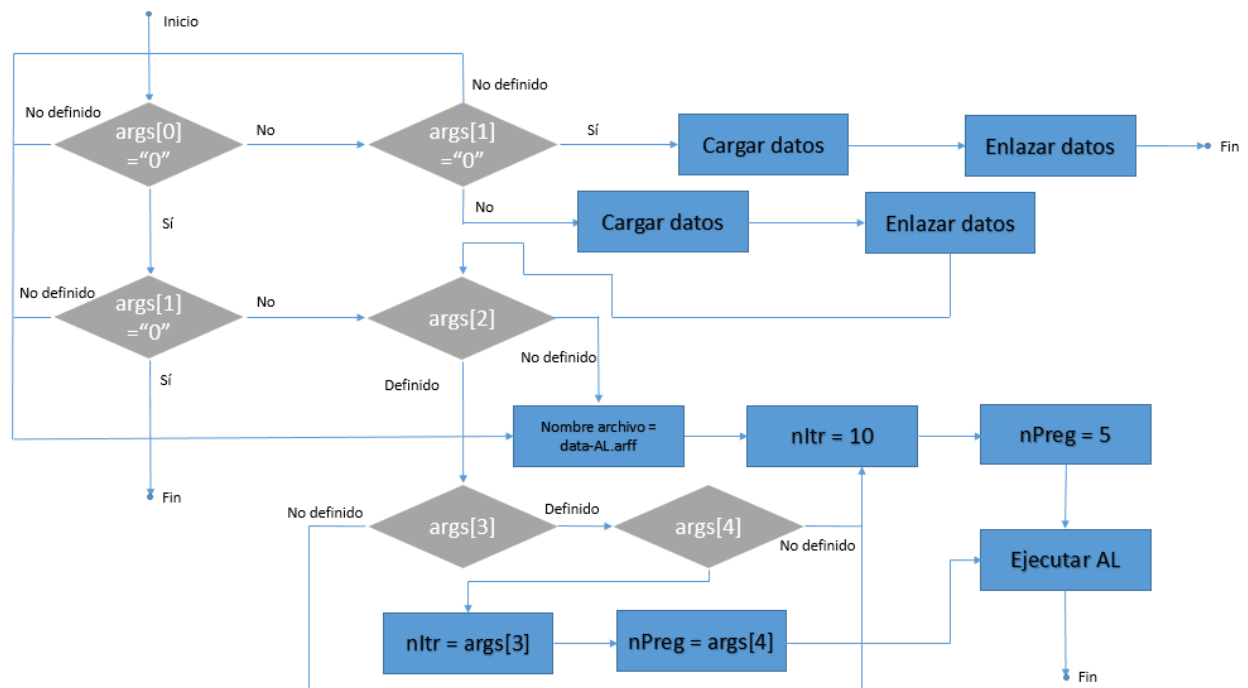


Figura 6.5: Diagrama de flujo del sistema decisor experto.

figurando en la raíz de la carpeta del proyecto junto con el archivo *p-merge-t.arff*, que contiene la señal de dolor. El módulo se encuentra implementado en el proyecto Java, figurando en el paquete *principal* en una clase de nombre *ExpSysAL.java* que hemos ejecutado en un entorno Java cualquiera. El entrenamiento con el sistema experto requiere un proceso previo de enlazamiento de datos bastante largo en procesadores comunes, del que hacemos abstracción, así como en el caso de la ejecución del ANFIS, ya que en el sistema final funcionará corriendo en un servidor online. El proceso es opcional y puede eliminarse mediante el primer argumento del programa, como podemos ver en su diagrama de flujo(ver Figura 6.5). Si ponemos 0 no se ejecutará, y si es cualquier otra cosa se ejecutará. Se requiere también definir si acto seguido se ejecutará o no el AL mediante el segundo argumento, siguiendo la misma política de decisión. Si no introducimos argumentos, sólo se ejecutará el experimento y la parte de enlazado de datos se ignorará por defecto. Permitimos el largo tiempo de enlazado de datos porque nos importan principalmente los resultados del entrenamiento y no tanto su eficiencia. Tras terminar el proceso de enlazado de datos, hemos obtenido un archivo llamado *data-AL.arff*, o del nombre que queramos, especificado en el tercer argumento, como podemos ver en el diagrama de flujo(ver Figura 6.5). El archivo resultante son los datos, que dividiremos en un conjunto de entrenamiento y en un conjunto de test siguiendo el criterio visto en el Capítulo 4.

La introducción de la información en los diversos módulos WEKA puede dar errores con las etiquetas temporales, en concreto *201703260200* y el resto de minutos de esa hora. Sospechamos que se trata de alguna limitación interna de la librería. Para solucionarlo hay que eliminar todos

los minutos de dicha hora, los datos fundamentales no se ven alterados.

El sistema experto se ejecuta, requiriendo al usuario las diversas instancias a clasificar en base a los criterios expuestos en el Capítulo 4. Dichas clasificaciones son introducidas para el entrenamiento progresivo del sistema, que construye el modelo final que aplica sobre los datos del archivo de test. Tras finalizar el entrenamiento, nos ha generado las diversas estadísticas calculadas.

Hemos realizado 5 experimentos en total, en los cuales se han realizado:

- 10 iteraciones con 5 preguntas por iteración.
- 50 iteraciones con 10 preguntas por iteración.
- Lo mismo que el primer punto, pero sin síntomas.
- Lo mismo que el primer punto, pero sin actividad.
- Lo mismo que el primer punto, pero sin síntomas ni actividad.

Lo que hemos procurado con los dos primeros experimentos es estudiar los efectos del algoritmo utilizado, mientras que los restantes han buscado aislar el efecto que las variables incorporadas a la alarma de dolor biomédica y ambiental hayan podido tener en el proceso de aprendizaje.

Podemos modificar el número de iteraciones y preguntas con los argumentos cuarto y quinto, como podemos ver en el diagrama de flujo del sistema (ver Figura 6.5). Se requiere poner el nombre del archivo y los datos previos sobre el comportamiento del programa. En caso contrario, el programa realizará por defecto 5 preguntas y 10 iteraciones.

Capítulo 7

Resultados

En el siguiente capítulo mostramos los datos experimentales que hemos obtenido del experimento, siguiendo los pasos referidos en la Sección 6.2.

La presentación de los resultados la hemos desarrollado en función del patrón que vimos en el Capítulo 5, mediante una representación de los resultados basada en tablas, desgranando, en base a ellas, explicaciones de las diversas causas del comportamiento del sistema.

7.1. Resultados del decisor biomédico

Comenzamos exponiendo los resultados obtenidos del sistema decisor biomédico que vimos en la Sección 5.2. La tabla (ver Tabla 7.1) alberga tanto los minutos de retardo como los valores generados por la matriz de confusión, siendo:

- TP: Verdaderos positivos.
- FP: Falsos positivos.
- TN: Verdaderos negativos.
- FN: Falsos negativos.

Observamos específicamente la columna *Minutos de retardo*. Entre la variación de los ANFIS de la primera mitad y los de la segunda hay un gran salto. Esto es debido a que el primer conjunto se está testeando sobre la curva trivial P/NP, mientras que el otro conjunto, salvo contadas excepciones y algunas migrañas aisladas, se está testeando sobre la curva generada por la caja negra en forma de campana gaussiana. Esta situación hace que las métricas se encuentren fuertemente sesgadas en ambos casos y hay que tenerla en cuenta. La desviación típica es $\sigma(fits) = 128.2766$. El mayor retraso corresponde al ANFIS 18, con 26 minutos y medio, y el mayor adelanto lo tiene el ANFIS 16, que responde con 527 minutos de adelanto, más de ocho horas.

Esta gran disparidad de retardos puede deberse a la naturaleza del método de entrenamiento seleccionado. El método de dejar uno fuera no tiene en cuenta la distribución de la información entre las particiones y hace que estas se comporten de forma muy desigual.

Sin embargo, como podemos ver en las dos últimas entradas, al seleccionar los dos ANFIS de ambas mitades, el segundo y el decimoctavo, y entrenarlos para el último test, al entrenarse

	Minutos de retardo	TP	FP	TN	FN
ANFIS 1	3.4	1160	16	8078	26
ANFIS 2	3.5	1352	16	3532	20
ANFIS 3	4.2857	3561	24	6745	50
ANFIS 4	3.2	1115	22	11738	25
ANFIS 5	4	7494	10	3084	32
ANFIS 6	4.25	637	15	13364	24
ANFIS 7	1.5	2719	24	9217	40
ANFIS 8	2.9167	3475	48	6489	68
ANFIS 9	1.875	2771	29	27401	39
ANFIS 10	-1.2	288	10	211	12450
ANFIS 11	-15.6667	3448	100	47939	1793
ANFIS 12	-53.1429	3882	551	71686	201
ANFIS 13	-9.2857	2076	158	72542	104
ANFIS 14	12.75	2574	4	37802	2820
ANFIS 15	-158.25	5834	633	36412	1761
ANFIS 16	-527	9944	3096	42566	554
ANFIS 17	-2.6667	469	39	23882	90
ANFIS 18	26.3333	868	17	58038	117
Test 1	1.3846	10392	101	86320	387
Test 2	1.3846	10392	101	86320	387

Tabla 7.1: Resultados del decisor biomédico.

Evento	TPR	TPV	F-Valor
Entrenamiento (10-CV)	0.91	0.84	0.88
Test	0.91	0.84	0.88

Tabla 7.2: Resultados del predictor ambiental con pacientes del 2014 y del 2017.

con el conjunto de entrenamiento completo las diferencias se atenúan, pues ambos test nos dan resultados idénticos en cuanto a retraso en minutos. Conseguimos que se atrase con una distancia muy pequeña de la alarma de la migraña original, por lo que predice de forma satisfactoria la crisis de migraña.

En promedio, por la entrada de la tabla *FP*, el sistema se inventa poca respuesta de dolor, si bien es mucho más acusada en la segunda mitad. La razón es la diferencia del tipo de información tratada por el primer grupo y el segundo grupo: dolor/no dolor, y curva de intensidad.

A nivel global podemos ver que el sistema es bueno y nos dará una respuesta satisfactoria. Sin embargo, no hemos realizado aquí elucubraciones sobre su conveniencia, ya que esto depende de los intereses particulares de la práctica médica que, particularmente en este aspecto, se encuentra en disputa.

7.2. Resultados del predictor ambiental

Procedamos a la exposición de los resultados obtenidos del predictor ambiental. Nos hemos centrado en las dos matrices de confusión (ver Tabla 7.3 y ver Tabla 7.4), pues han presentado problemas graves sobre el proceso de aprendizaje en el caso de los dos conjuntos de pacientes

	NP	A	P
NP	6565	0	0
A	20	0	0
P	555	0	0

Tabla 7.3: Matriz de confusión de entrenamiento del sistema predictor ambiental con pacientes del 2014 y del 2017.

	NP	A	P
NP	2814	0	0
A	9	0	0
P	238	0	0

Tabla 7.4: Matriz de confusión de test del sistema predictor ambiental con pacientes del 2014 y del 2017.

unidos. Lo que podemos extraer de ambas tablas es que el sistema es muy capaz de detectar eventos de no dolor, pero los eventos de dolor y de aura los pierde de vista en todos los casos, reduciéndolos a eventos de no dolor.

Inferimos que la hipótesis entrenada responderá que nunca hay dolor ni aura, por lo que no aportará información significativa al sistema experto.

Nos encontramos con que el F-Valor (ver Tabla 7.2) es más bajo que el TPR, siendo 0.88 y 0.91. Por lo que, haciendo abstracción de las frecuencias de aparición de las clases, se ha determinado que la predicción obtenida es mala para elementos de menor frecuencia de aparición, como las auras o el dolor.

En entrenamientos previos al experimento, el conjunto de los pacientes del año 2014 dio unos resultados parcialmente opuestos a estos. Sin embargo, los datos utilizados estaban contaminados con información de carácter biomédico, como intensidad de dolor global y administración o no de medicación al paciente. En esta ocasión, estos datos se han desechado y se han dejado sólo los referidos a aspectos ambientales.

Procedamos al examen de los experimentos realizados con los conjuntos separados por año.

Si nos fijamos en las matrices de confusión de los pacientes del 2014 (ver Tabla 7.6 y ver Tabla 7.6), veremos que sucede algo muy similar a la matriz del entrenamiento en bloque, por lo que en este caso no ganamos nada en cuanto a calidad de aprendizaje. Confunde constantemente auras y pródromos, tratándolos como dolor. Esta falta de ganancia es atestiguada por la comparación del F-Valor (ver Tabla 7.5) con respecto al TPR, muy inferior. La situación es similar al entrenamiento en bloque.

La situación cambia parcialmente si observamos los resultados generados por las matrices de confusión de los pacientes del año 2017 (ver Tabla 7.9 y Tabla 7.10). Ambas nos indican que el

Evento	TPR	TPV	F-Valor
Entrenamiento (10-CV)	0.80	0.64	0.71
Test	0.79	0.63	0.70

Tabla 7.5: Resultados del predictor ambiental con pacientes del 2014.

	NP	A	P
NP	792	3	0
A	17	1	0
P	177	0	0

Tabla 7.6: Matriz de confusión de entrenamiento del sistema predictor ambiental con pacientes del 2014.

	NP	A	P
NP	341	0	0
A	9	0	0
P	77	0	0

Tabla 7.7: Matriz de confusión de test del sistema predictor ambiental con pacientes del 2014.

modelo está discriminando de forma unilateral los datos de aura debido a su escasez. Sin embargo, en comparación con los pacientes del año 2017, las predicciones de dolor son bastante más sólidas. En la celda de falsos negativos de no dolor se encuentra en cada caso más del 50 % de los datos de dolor. El predictor ambiental en este caso es poco preciso en cuanto a funcionamiento real, pero sí detecta cierta estructuralidad. El F-Valor (ver Tabla 7.8) se acerca más a la información emitida por el TPR, teniendo como valores 0.94 y 0.95.

Podemos inferir de aquí varias cosas:

- Que los dos sensores han repercutido en la calidad de las masas de entrenamiento para la construcción del modelo.
- Que las diferencias de fechas entre los dos años, debido a que en el proceso de aprendizaje se han incorporado, pueden ser influyentes.

Los datos discretizados del conjunto de pacientes del 2017 dieron mayor estructuración que los enriquecidos con decimales del conjunto del 2014, al contrario de lo que esperábamos. Sin embargo, esto puede ser engañoso en cuanto al módulo entrenado con datos sencillos, los del 2017, pues los datos no contemplan las precipitaciones (ver Sección 6.1), algo irreal para las fechas que los datos han cubierto, y pierde decimales por lo que adolece de mucha imprecisión.

Inferimos que en el entrenamiento las fechas repercuten de forma activa en el proceso de aprendizaje. Debido a la disparidad de datos en fechas similares para los dos años, puede haber procedido asimilando fechas entre ambos, lo que ha hecho que la asociación posible del año 2017 se perdiese.

Si bien esta prueba concreta no pone en entredicho la posibilidad de predecir migrañas en sí, sí diremos que, para nuestro modelo y contando con los datos que tenemos, tal posibilidad queda en entredicho.

Evento	TPR	TPV	F-Valor
Entrenamiento (10-CV)	0.95	0.94	0.94
Test	0.94	0.94	0.93

Tabla 7.8: Resultados del predictor ambiental con pacientes del 2017.

	NP	A	P
NP	5726	0	44
A	1	0	0
P	257	0	120

Tabla 7.9: Matriz de confusión de entrenamiento del sistema predictor ambiental con pacientes del 2017.

	NP	A	P
NP	2466	0	7
A	1	0	0
P	130	0	32

Tabla 7.10: Matriz de confusión de test del sistema predictor ambiental con pacientes del 2017.

7.3. Resultados del sistema experto

Procedamos a exponer los resultados que hemos obtenido del sistema experto en los diversos experimentos que hemos planteado en la Sección 6.2. Contamos de partida con que la influencia del sistema predictor ambiental es nula porque hemos utilizado el modelo de predicción global, y no los particulares para cada conjunto. Los dos primeros experimentos van encaminados a comprobar el comportamiento del algoritmo Active Learning mediante un método diferencial, por lo que hemos opuesto en el análisis a ambos. El resto de experimentos han hecho abstracción sucesivamente de las diversas variables incorporadas para discernir si de ellas hay algún rastro de incidencia.

En el primer experimento (ver Tabla 7.11), nos hemos encontrado con que el sistema experto consigue una muy satisfactoria tasa de predicción en base a unos valores muy igualados tanto de TPR, PPV y F-Valor, que sólo se diferencian en los decimales menos significativos. A partir de la tercera iteración, la mejora se estanca, quedándose detenida para cada valor en un 92 % de tasa de corrección del test. El algoritmo AL, al realizar las preguntas, las propone principalmente para valores NP y P que se encuentran en zonas de frontera de decisión para la alarma biomédica (0.5-0.7). En estos casos nosotros hemos aportado mecánicamente la respuesta de NP y P como información a incorporar.

El segundo experimento no ha diferido mucho en comportamiento salvo en que en ocasiones ha preguntado por instantes asociados a pródromo/aura. En total las preguntas de este estilo se dieron en dos ocasiones, en la iteración 13, en la 18 y en la 27, y se dieron con tres valores de alarma frontera, 0,7/0.69/0,69. Si observamos la tabla (ver Tabla 7.12), podemos ver que no hay una gran variación entre las métricas de las dos iteraciones. Estas incorporaciones han pasado desapercibidas, y es posible se hayan subsumido en el conjunto de las 10 preguntas de cada iteración. Habremos aclarado el límite en pródromos, pero no repercute a nivel global.

Con respecto al primer experimento, el aumento de la masa de datos etiquetada ha mejorado la calidad del aprendizaje como queda reflejado en ambas tablas (ver Tabla 7.11 y Tabla 7.12), pero de un modo muy ínfimo.

El criterio que hemos seguido para clasificar los datos de pródromo en todos los casos en que

Evento	TPR	TPV	F-Valor	Etiquetados	No etiquetados
Iteración 1	0.9989	0.9989	0.9989	41333	419661
Iteración 2	0.9990	0.9990	0.9990	41338	419656
Iteración 3	0.9992	0.9992	0.9992	41343	419651
Iteración 4	0.9992	0.9992	0.9992	41348	419646
Iteración 5	0.9992	0.9992	0.9992	41353	419641
Iteración 6	0.9992	0.9992	0.9992	41358	419636
Iteración 7	0.9992	0.9992	0.9992	41363	419631
Iteración 8	0.9992	0.9992	0.9992	41368	419626
Iteración 9	0.9992	0.9992	0.9992	41373	419621
Iteración 10	0.9992	0.9992	0.9992	41378	419616

Tabla 7.11: Resultados del sistema experto en el primer experimento.

han aparecido han sido: si se encuentran entre 0 y 0.3-0.4 introducimos NP, y si se encuentran entre 0.4-0.5 y 1 introducimos P. Así hemos buscado influir en el proceso de definición de pródromo.

Proseguimos con la exposición de los tres últimos experimentos, en los cuales hemos aislado la alarma biomédica de los síntomas y de los datos de actividad.

El primer experimento de los tres aísla los resultados de las posibles influencias de los síntomas. Comparando las tablas (ver Tabla 7.11 y Tabla 7.12) ambas son idénticas. Podemos inferir que la influencia de los síntomas es ínfima.

Continuamos exponiendo los resultados del aislamiento del conjunto de datos de los datos de actividad. Sus respectivas tablas (ver Tabla 7.11 y ver Tabla 7.14) reflejan que la actividad sí tiene un efecto sobre la predicción de la migraña más acusado que los síntomas. Inferimos que la disposición de los datos de actividad, asociados a la temporalidad del dolor de la migraña, ha hecho que el sistema lo considere estructural.

El último experimento, que aísla el efecto tanto de los pródromos como de los datos de actividad, como vemos en la Tabla 7.15, nos ha unos resultados casi idénticos a los de la Tabla 7.14, algo peores, achacable o a la interacción humano-máquina existente en el proceso de aprendizaje o al efecto oculto de la eliminación de los pródromos en la Tabla 7.13.

Intuimos que la alta calidad de la respuesta según las métricas obtenidas mediante Hold Out pueda deberse a que el sistema depende de la alarma biomédica, que salvo un pequeño retardo es muy similar a la alarma que se entrena en el sistema experto. Vemos con seguridad que el sistema decisor biomédico en solitario sí es útil, y que el uso del resto de información apenas ayuda en el sistema experto final.

Evento	TPR	TPV	F-Valor	Etiquetados	No etiquetados
Iteración 1	0.9989	0.9989	0.9989	41333	419661
Iteración 2	0.9990	0.9990	0.9990	41343	419651
Iteración 3	0.9992	0.9992	0.9992	41353	419641
Iteración 4	0.9992	0.9992	0.9992	41363	419631
Iteración 5	0.9992	0.9992	0.9992	41373	419621
Iteración 6	0.9992	0.9992	0.9992	41383	419611
Iteración 7	0.9992	0.9992	0.9992	41393	419601
Iteración 8	0.9992	0.9992	0.9992	41403	419591
Iteración 9	0.9991	0.9991	0.9991	41413	419581
Iteración 10	0.9992	0.9992	0.9992	41423	419571
Iteración 11	0.9992	0.9992	0.9992	41433	419561
Iteración 12	0.9992	0.9992	0.9992	41443	419551
Iteración 13	0.9992	0.9992	0.9992	41453	419541
Iteración 14	0.9992	0.9992	0.9992	41463	419531
Iteración 15	0.9992	0.9992	0.9992	41473	419521
Iteración 16	0.9993	0.9993	0.9993	41483	419511
Iteración 17	0.9993	0.9993	0.9993	41493	419501
Iteración 18	0.9992	0.9992	0.9992	41503	419491
Iteración 19	0.9992	0.9992	0.9992	41513	419481
Iteración 20	0.9994	0.9994	0.9994	41523	419471
Iteración 21	0.9993	0.9993	0.9993	41533	419461
Iteración 22	0.9993	0.9993	0.9993	41543	419451
Iteración 23	0.9993	0.9993	0.9993	41553	419441
Iteración 24	0.9993	0.9993	0.9993	41563	419431
Iteración 25	0.9993	0.9993	0.9993	41573	419421
Iteración 26	0.9994	0.9994	0.9994	41583	419411
Iteración 27	0.9994	0.9994	0.9994	41593	419401
Iteración 28	0.9993	0.9993	0.9993	41603	419391
Iteración 29	0.9994	0.9994	0.9994	41613	419381
Iteración 30	0.9994	0.9994	0.9994	41623	419371
Iteración 31	0.9994	0.9994	0.9994	41633	419361
Iteración 32	0.9994	0.9994	0.9994	41643	419351
Iteración 33	0.9994	0.9994	0.9994	41653	419341
Iteración 34	0.9994	0.9994	0.9994	41663	419331
Iteración 35	0.9994	0.9994	0.9994	41673	419321
Iteración 36	0.9993	0.9993	0.9993	41683	419311
Iteración 37	0.9994	0.9994	0.9994	41693	419301
Iteración 38	0.9994	0.9994	0.9994	41703	419291
Iteración 39	0.9994	0.9994	0.9994	41713	419281
Iteración 40	0.9994	0.9994	0.9994	41723	419271
Iteración 41	0.9994	0.9994	0.9994	41733	419261
Iteración 42	0.9994	0.9994	0.9994	41743	419251
Iteración 43	0.9994	0.9994	0.9994	41753	419241
Iteración 44	0.9994	0.9994	0.9994	41763	419231
Iteración 45	0.9994	0.9994	0.9994	41773	419221
Iteración 46	0.9994	0.9994	0.9994	41783	419211
Iteración 47	0.9994	0.9994	0.9994	41793	419201
Iteración 48	0.9994	0.9994	0.9994	41803	419191
Iteración 49	0.9994	0.9994	0.9994	41813	419181
Iteración 50	0.9994	0.9994	0.9994	41823	419171

Tabla 7.12: Resultados del sistema experto en el segundo experimento.

Evento	TPR	TPV	F-Valor	Etiquetados	No etiquetados
Iteración 1	0.9989	0.9989	0.9989	41333	419661
Iteración 2	0.9990	0.9990	0.9990	41338	419656
Iteración 3	0.9992	0.9992	0.9992	41343	419651
Iteración 4	0.9992	0.9992	0.9992	41348	419646
Iteración 5	0.9992	0.9992	0.9992	41353	419641
Iteración 6	0.9992	0.9992	0.9992	41358	419636
Iteración 7	0.9992	0.9992	0.9992	41363	419631
Iteración 8	0.9992	0.9992	0.9992	41368	419626
Iteración 9	0.9992	0.9992	0.9992	41373	419621
Iteración 10	0.9992	0.9992	0.9992	41378	419616

Tabla 7.13: Resultados del sistema experto en el tercer experimento.

Evento	TPR	TPV	F-Valor	Etiquetados	No etiquetados
Iteración 1	0.9925	0.9925	0.9924	41333	419661
Iteración 2	0.9925	0.9925	0.9924	41338	419656
Iteración 3	0.9925	0.9925	0.9924	41343	419651
Iteración 4	0.9927	0.9927	0.9926	41348	419646
Iteración 5	0.9927	0.9927	0.9926	41353	419641
Iteración 6	0.9927	0.9927	0.9926	41358	419636
Iteración 7	0.9927	0.9927	0.9926	41363	419631
Iteración 8	0.9927	0.9927	0.9926	41368	419626
Iteración 9	0.9927	0.9927	0.9926	41373	419621
Iteración 10	0.9927	0.9927	0.9926	41378	419616

Tabla 7.14: Resultados del sistema experto en el cuarto experimento.

Evento	TPR	TPV	F-Valor	Etiquetados	No etiquetados
Iteración 1	0.9925	0.9925	0.9924	41333	419661
Iteración 2	0.9925	0.9925	0.9924	41338	419656
Iteración 3	0.9925	0.9925	0.9924	41343	419651
Iteración 4	0.9926	0.9927	0.9926	41348	419646
Iteración 5	0.9926	0.9927	0.9926	41353	419641
Iteración 6	0.9926	0.9927	0.9926	41358	419636
Iteración 7	0.9926	0.9927	0.9926	41363	419631
Iteración 8	0.9926	0.9927	0.9926	41368	419626
Iteración 9	0.9927	0.9927	0.9926	41373	419621
Iteración 10	0.9926	0.9927	0.9926	41378	419616

Tabla 7.15: Resultados del sistema experto en el quinto experimento.

Capítulo 8

Conclusiones y trabajo futuro

Buscamos predecir crisis de migraña mediante técnicas de aprendizaje automático. A día de hoy, han surgido diversos enfoques en el campo de la predicción de migrañas que han respondido al problema.

Uno ha dado muy buenos resultados. Se basa en el entrenamiento de un modelo de aprendizaje automático construido mediante variables hemodinámicas y la intensidad del dolor sufrida por el paciente [1]. Como criterio de decisión se empleó un módulo muy sencillo. Se trata de un umbral que detecta si hay migraña [1]. Sin embargo, debido a los posibles errores en la señal derivados de su uso en la vida real, el umbral puede dar falsos positivos al ser superado sin haber episodios de migraña [1]. Se hace necesaria la mejora de este criterio de decisión.

En este trabajo hemos desarrollado un sistema decisor experto que actúa como sistema de apoyo a decisión de ese sistema predictor de migrañas basado en variables hemodinámicas y en datos de intensidad de dolor, productor de variables biomédicas, siguiendo el criterio explicado en la Sección 5.1. El sistema decisor experto integra además información periférica de tipo ambiental, prodrómica y de actividad.

El propósito del sistema (ver Sección 1.2) es embeberlo en un dispositivo ambulatorio que permita la extracción y tratamiento de datos acerca de la migraña, y su posterior predicción en un entorno real bajo monitorización médica.

Este sistema consta de de las siguientes partes, sintetizadas en la Figura 5.1:

- Un primer módulo productor de variables biomédicas que figura en nuestro proyecto como caja negra.
- Un módulo que implementa un sistema decisor biomédico previo al sistema decisor experto final.
- Un módulo que implementa un sistema predictor ambiental.
- Una base de datos con información de pródromo y actividad.
- Un módulo que implementa el sistema experto final.

Los resultados han mostrado que:

- El sistema decisor biomédico previo al sistema decisor experto ha conseguido detectar migrañas con un retardo de 1.38 minutos en promedio, suponiendo una muy buena capacidad de detección que repercute significativamente en la robustez del sistema experto final (ver Sección 7.1).
- El sistema predictor ambiental ha fallado al entrenar el modelo como un conjunto unitario para los dos conjuntos de pacientes, devolviendo sólo información de No Dolor en su salida. Pudimos ver que por separado podrían habernos dado un éxito parcial, aunque sospechamos que engañoso (ver Sección 7.2).
- En la integración de lo obtenido mediante los dos sistemas anteriores en un sistema decisor experto final mayor, la influencia de los datos ambientales según el enfoque utilizado es nula, la de los datos prodrómicos es muy débil y la de los datos de actividad de los pacientes es algo más relevante, comandando la robustez de su respuesta la alarma del sistema decisor previo basado en lógica difusa. Salvo para aumentar ligeramente la precisión, según lo que deja traslucir nuestros experimentos, la información periférica no nos sirve de mucho (ver Sección 7.3).

Se puede concluir que el módulo de detección inicial es el más (y quizá el único) adecuado para detectar adecuadamente crisis migrañosas en el dispositivo ambulatorio propuesto en [1]. Para mejorar los resultados proponemos diversos puntos como líneas de trabajo futuras.

Para el sistema decisor biomédico, proponemos:

- En primer lugar, trabajar en la calidad de los datos obtenidos para entrenamientos futuros. Su disparidad nos infundó cierta desconfianza con respecto al funcionamiento del sistema.
- En segundo lugar, incorporar en la intensidad de dolor acontecimientos previos que permitan perfilar su forma de un modo más preciso, sean las molestias de los síntomas del pródromo o del aura, dando así más base objetiva a la construcción de las debidas curvas de ser posible.
- En tercer lugar, realizar una implementación más general del detector ANFIS que sea capaz de aleatorizar los datos de los pacientes e introducir un número personalizado de pacientes.

Para el sistema predictor ambiental:

- En primer lugar, barajar la idea utilizada por el proyecto Migraine Alert (ver Sección 3.2) adaptada a nuestro contexto, y entrenar un modelo ambiental para cada paciente, teniendo en cuenta el ambiente en el que se mueva en su día a día. La posibilidad de un modelo general queda desechada, lo que puede estar en cierto modo en coherencia con lo dicho en la Sección 3.1. Sin embargo, dada su evidencia científica, no consideramos conveniente desechar completamente este enfoque.
- En segundo lugar, es necesario unificar los sensores de extracción de datos para evitar la repercusión de sus diversas calidades en los resultados.

Y para el sistema decisor experto final:

- En primer lugar, de seguir utilizando la técnica Active Learning en futuros casos, aunque fuese como apoyo experimental, la construcción de una clase HoldOut propia, que permita mayor flexibilidad a la hora de extraer datos del sistema experto y obtener una clasificación de los datos o el clasificador final. Se han explorado todas las vías que la librería empleada ofrece de forma nativa, pero el tratamiento de datos interno de la clase HoldOut original hace que sean inútiles si no es ampliando sus métodos.
- En segundo lugar, también es importante que, para evitar la alta latencia de enlazamiento de datos, se emplee una base de datos real que nos permita eliminar la alta complejidad computacional del método simple que hemos implementado para no complicar mucho el sistema.

Capítulo 9

Conclusions and future work

The aim of this work is to predict migraine crisis with machine learning techniques. Nowadays, there are diverse approaches in the field of migraine prediction that have tackled this problem.

One of them has offered good results. It is based on the training of a machine learning model built with hemodynamic variables and the intensity of the pain suffered by the patient. A simple module was used as decision criteria. It detects if a migraine occur with a threshold. But, because of the possible errors in the signal in the real live use, the threshold could give false positives if is reached without migraine episodes. Thus, the improvement of this decision criteria is necessary.

In this work we have developed an expert decision system that acts as a SSD for that migraine predictor system based on hemodynamic variables and intensity of pain data. The expert decision system also integrates as peripheral information environmental, prodromic and activity data.

The purpose of the system is to introduce it in an ambulatory device that allows the extraction and treatment of data about migraine, and its later prediction in a real environment under medical monitoring.

This system consists of these parts:

- A first module producer of biomedical variables included in our project as a black box.
- A module that implements a previous biomedical decision system for the final expert decision system.
- A module that implements an environmental predictor system.
- A database with prodromic and activity data.
- A module that implements a final expert system.

The results showed that:

- The previous biomedical decision system for the the expert decision system have achieved the detection of migraines with a delay of 1.38 minutes on average, implying a very well performance on detection that affects significantly on the robust response of the final expert system.

- The environmental predictor system has failed when we tried to train the model as an unitary set for the two sets of patients, returning only No Pain information on its output. We could see that separately they could give a partial success, but we suspect that is deceptive.
- On the integration of the information obtained with the previous systems in a mayor expert decision system, the environmental data influence with the implemented approach is null, the influence of prodromic data is very weak and activity data is a bit more pronounced, commanding the robustness of response the alarm of the previous decision system based on fuzzy logic. So, except to slightly increase the accuracy of the system, as our experiments show, the peripheral information is not very useful.

We could conclude that the first decision module is the more (and maybe the unique) adequate for the decision of a migraine crisis event. In order to improve the results we propose some points as future work lines.

For the biomedical decision system:

- In first place, work on the quality of obtained data for further trainings, because its disparity gave us some mistrust with its performance.
- In second place, include in the pain intensity variable previous events that allow outline its shape in a more precise manner, the annoyance of prodromic symptoms or the aura events, giving more objective base for the construction of the curves.
- In third place, make a more general implementation of the ANFIS detector that could be able to randomize the patients data and to introduce a customized number of patients.

For the environmental predictor system:

- In first place, to raise the idea used by the Migraine Alert project adapted in our context, and train an environmental model for each patient. The possibility of a general model is discarded, which it could be in coherence in some manner with the things said in about migraine prediction with environmental data. However, given its scientific evidence, we will not consider conveniently to discard completely this approach.
- In second place, is needed to unify the data extraction sensors to avoid the negative influence of the diverse quality on their results.

For the expert decision system:

- In first place, if we will continue the use of Active Learning technique on future cases, even as an experimental support, develop an own HoldOut class that allows to us a greater flexibility in extraction of data from the expert system and in obtaining a classification of the data or the final classifier. We have explored all the ways that JCLAL offers natively, but the internal data handling of the original HoldOut class makes them useless without the expansion of its methods.

- In second place, is also important, in order to avoid the high latency of data binding, to use a real database that allows us to avoid the high computational complexity of the simple method that we have implemented so as not to complicate the system.

Bibliografía

- [1] J. Pagán, D. Orbe, M. Irene, A. Gago, M. Sobrado, J. L. Risco-Martín, J. V. Mora, J. M. Moya, and J. L. Ayala, “Robust and accurate modeling approaches for migraine per-patient prediction from ambulatory data,” *Sensors*, vol. 15, no. 7, pp. 15419–15442, 2015.
- [2] N. Giffin, L. Ruggiero, R. Lipton, S. Silberstein, J. Tvedskov, J. Olesen, J. Altman, P. J. Goadsby, and A. Macrae, “Premonitory symptoms in migraine an electronic diary study,” *Neurology*, vol. 60, no. 6, pp. 935–940, 2003.
- [3] J. Pagán, J. L. Risco-Martín, J. M. Moya, and J. L. Ayala, “Modeling methodology for the accurate and prompt prediction of symptomatic events in chronic diseases,” *Journal of biomedical informatics*, vol. 62, pp. 136–147, 2016.
- [4] D. I. Friedman and T. De Ver Dye, “Migraine and the environment,” *Headache: The Journal of Head and Face Pain*, vol. 49, no. 6, pp. 941–952, 2009.
- [5] A. Hundert, A. Huguet, P. McGrath, J. Stinson, and M. Wheaton, “Commercially available mobile phone headache diary apps: A systematic review. jmir mhealth uhealth. 2014; 2 (3): e36. doi: 10.2196/mhealth. 3452.”
- [6] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.
- [7] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews],” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.
- [8] B. Settles, “Active learning,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.
- [9] I. Güler and E. D. Übeyli, “Adaptive neuro-fuzzy inference system for classification of eeg signals using wavelet coefficients,” *Journal of neuroscience methods*, vol. 148, no. 2, pp. 113–121, 2005.
- [10] M. W. Gardner and S. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” *Atmospheric environment*, vol. 32, no. 14, pp. 2627–2636, 1998.

- [11] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, *et al.*, “Top 10 algorithms in data mining,” *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.
- [12] P. Van Overschee and B. De Moor, “N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems,” *Automatica*, vol. 30, no. 1, pp. 75–93, 1994.
- [13] J. Pagán, J. L. Risco-Martín, J. M. Moya, and J. L. Ayala, “Grammatical evolutionary techniques for prompt migraine prediction,” in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, pp. 973–980, ACM, 2016.
- [14] O. Reyes, E. Pérez, M. Del Carmen Rodríguez-Hernández, H. M. Fardoun, and S. Ventura, “Jclal: a java framework for active learning,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3271–3275, 2016.

Agradecimientos

A mis tutores, José Luis Risco Martín y Josué Pagán Ortiz, por darme apoyo y guía en la realización de este trabajo, sin cuyos consejos y atención constantes no habría sido posible.

A mi familia y amigos, quienes me han dado apoyo moral tanto en los últimos días del desarrollo de este proyecto como en lo que respecta al resto de facetas de mi vida.

Autorización de difusión

Autorización para la difusión del Trabajo Fin de Grado y su depósito en el Repositorio Institucional E-Prints Complutense

Los abajo firmantes, alumno y tutor del Trabajo Fin de Grado (TFG) en Ingeniería de Computadores de la Facultad de Informática, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor, el Trabajo Fin de Grado (TFG) cuyos datos se detallan a continuación. Así mismo autorizan a la Universidad Complutense de Madrid a que sea depositado en acceso abierto en el repositorio institucional con el objeto de incrementar la difusión, uso e impacto del TFG en Internet y garantizar su preservación y acceso a largo plazo.

TÍTULO del TFG: Definición, desarrollo e implementación de un sistema experto de decisión para entornos de modelado e-health.

Curso académico: 2016/2017

Nombre del Alumno: **Fernando Viñas Martins.**

Tutor del TFG: **José Luis Risco Martín**, Departamento de Arquitectura de Computadores y Automática.

Tutor del TFG: **Josué Pagán Ortiz**, Departamento de Arquitectura de Computadores y Automática.

Firma del alumno

Firma del primer tutor

Firma del segundo tutor